Computer-Assisted Language Comparison: State of the Art

By comparing the languages of the world, we gain invaluable insights into human prehistory, predating the appearance of written records by thousands of years. The traditional methods for language comparison are based on manual data inspection. With more and more data available, they reach their practical limits. Computer applications, however, are not capable of replacing experts' experience and intuition. In a situation where computers cannot replace experts and experts do not have enough time to analyse the massive amounts of data, a new framework, neither completely computer-driven, nor ignorant of the help computers provide, becomes urgent. Such frameworks are well-established in biology and translation, where computational tools cannot provide the accuracy needed to arrive at convincing results, but do assist humans to digest large data sets. In this talk, we will illustrate what we consider the current state of the art of computer-assisted language comparison, by presenting a workflow that starts from raw data and leads up to a stage where sound correspondence patterns across multiple languages have been identified and can be readily presented, inspected, and discussed. We illustrate this workflow with help of a dataset on Hmong-Mien languages, which has so far not yet been analyzed in this way. Our illustration is furthermore accompanied by Python code and instructions on how to make use of additional web-based tools we developed, so that users can replicate our workflow or apply it for their own purposes.

1 Introduction

1.1 The Gap between Computational and Traditional Historical Linguistics

The proposal of new, fancy, and shiny quantitative methods applied to handle problems in historical linguistics has created a gap between what one could call "classical" approaches to historical language comparison and the "new and innovative" automatic approaches. Classical linguists are often skeptical of the new approaches, partly because the results differ from those achieved by classical methods (Anthony and Ringe 2015, Holm 2007), but also because the majority of the new approaches work in a black box fashion and do not allow inspecting the concrete findings in detail. Computational linguists, on the other hand, complain about classical historical linguists' lack of consistency when applying the classical methods.

1.2 Computer-Assisted Disciplines

The use of computer applications in historical linguistics is steadily increasing. With more and more data available, the classical methods reach their practical limits. At the same time, computer applications are not capable of replacing experts' experience and intuition, especially when data are sparse. If computers cannot replace experts and experts do not have enough time to analyse the massive amounts of data, a new framework is needed, neither completely computer-driven, nor ignorant of the assistance computers afford. Such computer-assisted frameworks are well-established in biology and translation. Current machine translation systems, for example, are efficient and consistent, but they are by no means accurate, and no one would use them in place of a trained expert. Trained experts, on the other hand, do not necessarily work consistently and efficiently. In order to enhance both the quality of machine translation and the efficiency and consistency of human translation, a new paradigm of computer-assisted translation has emerged (Barrachina et al. 2008: 3).

1.3 Computer-Assisted Language Comparison

Following the idea of computer-assisted frameworks in translation and biology, a framework for computer-assisted language comparison (CALC) could be the key to reconcile classical and computational ap-

proaches in historical linguistics. Computational approaches may still not be able to compete with human experts, but when used to pre-process the data with human experts systematically correcting the results, they can drastically increase both the efficiency and the consistency of the classical comparative method.

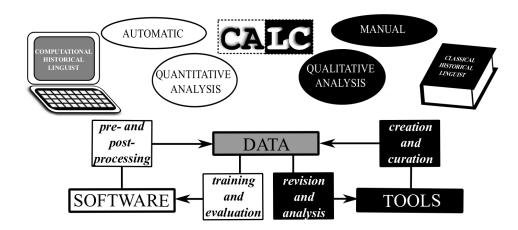


Figure 1: Basic idea of data managment within the CALC framework.

The basic idea behind computer-assisted as opposed to computer-based language comparison is to allow scholars to do qualitative and quantitative research are done at the same time. In order to allow scholars to do this, data must always be available in machine- and human-readable form. Figure 1 shows a tentative workflow for the CALC framework, in which data is constantly passed back and forth between computational and classical linguists.

Three different aspects are essential for this workflow:

- (a) New software allows for the application of transparent methods which increase the accuracy and the application range of current methods and also treat the peculiarities of specific language families (like, e.g., Sino-Tibetan).
- (b) Interactive tools provide an interface between human and machine, allowing experts to correct errors and to inspect the automatically produced results in detail.
- (c) Specific data is used to test and train the software algorithms.

2 Workflows for Computer-Assisted Language Comparison

2.1 Overview

Our workflows for computer-assisted language comparison have so far been intensively tested on a small set of 8 Burmish languages, which we investigated in collaboration with Nathan W. Hill, who was responsible for the qualitative investigation of the data and for the common discussion of new computer-assisted methods which were then implemented by Johann-Mattis List (see Hill and List 2017 for an exemplary discussion of some of the new approaches). Our experience with the Burmish project by now allows us to set up a first workflow that starts from raw data and leads up to the explicit identification of correspondence patterns across multiple languages. At the moment, List and Hill develop the workflow further to account also for (semi)-automatic reconstructions, but in this talk, only the identification of correspondence patterns will be discussed.

2.2 Details of the Workflow

Our workflow currently comprises 5 different stages, in which we successively lift linguistic data from their raw form in which we can find them in wordlists and tables published in dictionaries and field-work notes, up to a level where correspondence patterns across cognate words have been automatically identified and can be qualitatively inspected by the scholar.

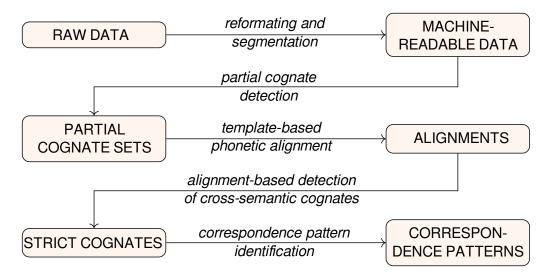


Figure 2: Current state-of-the-art workflow developed in collaboration of different research groups working in computer-assisted frameworks.

Although the workflow can be carried out almost completely without any manual intervention by a linguist, we emphasize that this workflow explicitly *allows* for expert intervention at *any* of the five stages. While, in our experience, specific care is required when lifting the data the first time to machine-readable format, it should further be noted that *all* steps of the workflow profit from human intervention, since none of the automatic methods currently available to us could spot all patterns in linguistic data without overor underestimating their importance for linguistic reconstruction.

Our workflow starts from raw data, including tabular data from fieldwork notes or data published in books and articles, which we re-organize and re-format in such a way that the data can be processed by our tools. Once we have machine-readable data, we can use methods for automatic cognate detection (List et al. 2016b) in order to infer partial cognates across the languages in our data. Having inferred cognates, we can now also align the data in the cognate sets. While we could use phonetic alignment approaches discussed in the literature (List 2014), we now use a new approach, based on phonotactic templates, which has the advantage of being much faster and accurate when dealing with alignments for South-East-Asian languages. Once having identified the alignments, we start to search automatically for cognates across different concepts. Since all automatic methods need to start searching for cognates within the same concept slot (otherwise, there would be too many false positives), our new method, which makes used of a systematic comparison of readily aligned cognate sets, systematically searches for cognates independent of their meaning. The improved, cross-semantic cognate sets, which are all readily aligned, have the specific property of being strict: no cognate set could compare two morphemes from the same language which would differ in their pronunciation. (List 2018) calls these cognate sets regular, but in discussions with Nathan Hill, we decided that regular is probably not the best term, as they can well be wrong, so we call them strict now. Once strict cognates have been identified, we use the new algorithm for the automatic inference of sound correspondence patterns across multiple languages by List (2019) to infer the correspondence patterns in the data.

In Section 3, we will provide detailed examples how all steps of the workflow interact, using a rela-

tively recent collection of linguistic data on Hmong-Mien languages (Chén 2012) for this purpose.

2.3 Materials and Methods for the Workflow Illustration

The data we use to illustrate our workflow in the next section was originally collected by Chén (ibid.), and later added in digital form to the Wiktionary project. Chén's collection of *frequent terms* (*chángyòng cíbiǎo* 常用词表, pp. 567-862) comprises 885 different concepts translated into 25 varieties of Hmong-Mien. In Figure 3, we contrast one exemplary page from Chéns book with the data as it has been prepared by the Wiktionary users. We can see that the data is essentially the same, but that the rows and columns of the tabular form have been swapped.

甲木	mer-ma	gaga	0000	ńa. ńa.	noijnoij	rs-aņu	ńeńe	ńeińei	ńehe	mei-mei-	0000		mar-her
前面	ptje24tei55zei33	tçi ⁵⁵ ði ³¹	tau ²⁴ nte ²¹	sö ¹³ plē ³¹	tçaŋ¹³mplæn⁵³	?i55tha11	kw44nə22	7zi35dauu31	tan118uu35	ton212ton22	tantilali	qa²¹tue⁵⁵	?aº³moŋ³
后面	p+je24tei55qsen33	tə ⁰² qwen ³¹	tau ²⁴ qaŋ ⁴³	sə ¹³ qə ⁵⁵	ntaŋ⁴⁴hua³²	?a ⁵⁰ qw ⁵⁰	kw44tei53	te	初二	初三	正月	taŋ²¹qãe⁴⁴	ta³¹tjeu⁴²
左边	p+je247a02zo33	?ə ⁰² ja ³¹	phag43lou13	be ₃₁ d9 ₀₃ te ₁₃₁	tçaŋ ¹³ lu ⁵³	saw ⁵⁵ fə ¹³	pa ⁴⁴ n _i ⁴²	ts T + 1		sen ³¹ pzi ³¹		tçi ²¹ tça ²¹	pu³¹qʰaŋ
右边	p*je24?a02nen24	?a∞ņe³s	phag43si33	pɛ³¹qəº²ntşə̄²²	tçaŋ¹²ҳaŋ²²	saw ⁵⁵ si ¹¹	pa⁴ta³¹				4a ²⁴ ma ³⁵	tçi ²¹ te ⁵⁵	pu³¹kwe³
上边	pə ⁰⁵ sa ³³	X9228033	§ou⁴⁴	5"e13Ze31	tcaŋ13pe11	?a55ga55	kw44lo53	ka 高 寨	sen³1 ?u³1	sen³1 pji³1	4a³s mass	xhæ44vei35	?aº³waŋ³
下边	tə ⁰² qhaŋ ⁵⁵	X955ti31	tce ³¹	kε ¹³ to ²²	ha ⁴⁴ to ¹³	pi ⁵⁵ dau ⁵³	kw44hag44	ta大南山	sa43 ?au43	sa43 pe43	Juass di of	xhæ44fæ55	?a°³tε⁴²ŋ,
里边	t902χwε55	χə ⁵⁵ η[οη ³¹	hou ⁵⁵ nţau ¹³	ntşö²²ntşö²²	[ə ⁴⁴	ndlo53	kw⁴⁴naŋ⁴²	hv ià 44	she 24 ? a 24	shê 24 p	zã 55 4a 42	xʰǣ⁴⁴tjuŋ⁴⁴	?aº3qɔ+2
外边	χο ³¹ ŋgwaŋ ²⁴	lu ³¹ tçoŋ ³¹	ntou ⁴⁴	şö ¹³ ntşo ⁴²	tçaŋ¹³ntɔ⁴⁴	Ghau ^{≥1}	kw44tei53	za 同 坡			4a 4a	xhæ44ku55	?aº³nɔ³¹ŋ
中间	pə ^{c5} ntuŋ ³¹	pa ^{co} ntog ^{a1}	ki²⁴nṭaŋ⁴³	kā³¹ntṣā²⁴	ntaŋ³²	pissntawss	ta ²⁵ ηţοŋ ²⁵	qē 宗 地	Sæn ²² ?ɔ ³ 2	sæn²² pæ³²	zan ^{s3} i as	qa²³tjuŋ⁴⁴	qoº3seŋ55
旁边	?i55p1je24	tə ^{cz} ηţαŋ ⁵⁵	nto13	qə ^{oz} mpö ^{oz}	tcag13ntog13	?a ⁵⁵ ntu ⁵³	do ₃₂ ca ₄₄	ta 石门坎	sie st ?a st	sie tsts1 ts	lust ? i st 4 i 11	qa³³xãe³³	?a⁰³jaŋ⁴⁴
今天	na³¹naŋ⁵⁵	ņa ³¹ ?noŋ ⁵⁵	ņo ⁴³ na ⁵⁵	ņē ²⁴ nā ¹³	noŋ ²³² na ⁴²	mu ³³ na ³³	tha53nen44	th	7m35 ne35 4a53	pu ³⁵ ne ³⁵ 4a ⁵³	da ⁵³ ?a ⁴⁴	tuë ³³ na ¹³	m³³ņei³5
明天	phau31na31	phe31ne31je35	tca ⁴⁴ ki ²¹	pe³¹ntşʰe⁴²	ha44nte13	pi ³¹ dzi ³¹	ci ³⁵ ŋe ³⁵	pi 腊乙坪		pa ne 40	1	tæ³³sew¹³	po³³ŋei³⁵
昨天	mo ⁵⁵ na ³¹	ta ³¹ ņin ⁵⁵ ņa ³¹	7a43nag13	qə ⁰² la ³¹ ŋə̄ ²⁴	η[0 ¹¹ noŋ ²³²	?a³¹naw³¹	n _i i ³¹ ņe ³⁵	小 章	7 u 33 mei 33 da 33	pu ³³ nei ⁵³ 4a ³³	tsen ⁵³ ye ¹³	nei**nag21	qaº³ŋei³5
后天	si ⁵⁵ na ³¹	si ⁵⁵ ņa ³¹	naŋ¹³ki²¹	ŋā²⁴tşo⁴²	noŋ²³²ηʈɔ⁴⁴	?a³¹naw³¹dzñi¹¹	?u³5pε⁴4ņe³5	70 基 薏	ne33 ?033	n 8 33 pi 33	4ha** ge35	çi44za ²¹	tçi ⁵⁵ ŋei ³⁵
上午		ti ⁵⁵ je ³¹	taŋ ⁵⁵ nau³¹tʂʰai³³		tæn42nte13	sey ⁵⁵ ntso ⁵⁵	te ³⁵ te ³⁵ ntu ³⁵	06 茎地渡	.29 . 26	. 24 _ 24	la tsen nje 33	kua²²tæ⁴	qa ^{o3} tei ⁵⁵
中午	tə ⁰² si ³¹ n ₂ a ²⁴ ji ³³	ηξοη ³¹ ηα ³¹	taŋ ⁵⁵ nau ³¹ şo ⁴⁴	ntşä²⁴ŋä²⁴	pje ¹³ soŋ ³⁵	nhau ³⁵ gu ³³	ņe ³⁵ ntu ³⁵	nd Ac a S	nei ?ou	7641		tjuŋ⁴⁴nei⁴⁴	ma ⁵⁵ n _e on
下午	tə ⁰² si ³¹ n ₂ a ²⁴ pjo ³³	n ₂ a31saŋ33	taŋ⁵⁵nau³¹mau⁴⁴	Shj24[823Sh842	he ³³ mo ³⁵	nhau ³⁵ şu ³³ tchau ³³	ņe ³⁵ ci ³⁵	m 羌 告	no13 2u13	no13 pai 13	dei 44 tig 24	tju ³³ sa ⁵³	qa ⁰³ ma ¹³
早晨	tei55ji33	tei55je31	şeu ⁵⁵ ntso ⁵⁵	ntşhe42	tæn42mpw ¹¹	sey55ntso55bhw11	qa**ntso**	ta 河 坝	nei 44 ? 3 44	nei 44 poi 49	la 33 7i 44	kua²¹sew¹³	qaº³tei55l
晚上	tə ⁰² muŋ²⁴	tə ⁰² moŋ³5	mau ⁴⁴ nto ³¹	mõ*²ŋqã55	tæn ⁴² mɔ ³⁵	mo33ndhu35	tci ³⁵ maŋ ⁵³	四 產 苦	heg ³⁵ ?ua ³⁵	heg ³⁵ po ³⁵	dast jei31	zi ³³ maŋ ³³	qaº³ma¹¹
半夜	?i55nduŋ²4muŋ²4	ηξοη ³¹ ηγοη ³⁵	taŋ¹³mau⁴⁴	?i²⁴tö³¹mö⁴²	toŋ ³³ mɔ ³⁵	daw³¹mo³³	tu ⁵³ mu ²²	?a 72 &				taŋ²¹maŋ³³	ma ⁵⁵ n _e on
初一	sen ³¹ ?i ²⁴	sen317i35	sa ⁴³ ?i ⁴³	shë247i24	sæn²²?ei³²	sie ⁵⁵ ?i ⁵⁵	?a⁴ŋe³> 1 a>³	?a 毛 坳	ta" ?ud"	ta ³³ Po ³⁵	tea" ne"	nei**?i**	heŋ³5jei³:
初二	sen317u31	sen317u31	sa43?au43	shëz47a24	sæn²²?ɔ³²	sie ⁵⁵ ?a ⁵⁵	?w³³ŋe³¾a³³	?u 21 百弄	Bag" 2533	Bag" Pe"	4u42 tsəŋ³³	nei**?>**	heŋ³5ʔua
初三	sen³¹pzi³¹	sen³¹pji³¹	sa ⁴³ pe ⁴³	shë24pe24	sæn²²pæ³²	sie ⁵⁵ ts1 ⁵⁵	pu ^{as} ņe ^{as} ła ^{sa}	pu a ப	50 ¹³ 1bi ²²	so ¹³ san ³³	lu ¹³ sen ³³	nei**poi**	heŋ³⁵po³
正月	fa ²⁴ ma ⁵⁵	ła³5ma55	lua ⁵⁵ fi ⁴⁴	χä ⁵⁵ fa⁴²	zaŋ ⁵³ li ³⁵	luss?jss4jss	fa53?a44	te.	Sai 33 ?de 33	sai ³³ pa ³³	tsi 33 40 44	la ³³ ?i ⁴⁴	ła ⁵⁵ jei ³¹
二月	ła²⁴zuŋ³¹	†a35zoŋ31	zaŋ³¹4i⁴⁴	nä²⁴ 1 a⁴²	na³²li³⁵	luss?ass+i11	tas=?w=s	五 瑶 麓	V F.			la ¹³ ?ɔ ⁴⁴	†a55?ua35
三月	ła²⁴tca³¹	4a35tea31	pe ⁴³ +i ⁴⁴	mî ³¹ †a ⁴²	men ¹¹ li ³⁵	lu ⁵⁵ ts1 ⁵⁵ fi ¹¹	+a33pu35	sa 巴 郡	tjheu ¹³ ŋ ⁵³	tjheu'³ son¹³	taken ¹³ nu ³	la ³³ poi ⁴⁴	+a35po33
四月	+a24qε31	+a35qe31	plou ⁴³ 4i ⁴⁴	zā554a42	zaŋ ⁵³ li ³⁵	lu ⁹⁵ t+au ⁹⁵ +i ²²	ła ⁵³ pzei ³⁵	fi ^s 优 诺	tshu ⁴⁴ Dji ³²	tshu ⁴⁴ deu ⁴⁴	tsen ⁴⁴ ŋje ²¹	la ³³ †3 ⁴⁴	+a ⁵⁵ tji ³⁵
五月	ła²⁴qlei³⁵	ta ³⁵ qlei ⁵⁵	tşi ⁴³ fi ⁴⁴	lë ²⁴ †a ⁴²	læn³²li³⁵	lu55pw554i21	+a53pza35	Pi 下水村	tsho 39 mii 35	tsho33 san 33	tsin22 ne33	la ³³ pja ⁴⁴	ła ^{ss} tja ^{ss}
六月	ła²4mpzi²4	+a³5mpji³5	tou++fi++	qe ²⁴ fa ⁴²	he ³² li ³⁵	lu ⁵⁵ t+aw ¹² +i ³³	łasstoss	to	Date 33.		-	la ³³ tju ³³	ła ^{ss} tęu ^{ss}
七月	ta ²⁴ nen ⁵⁵	ta ³⁵ nen ⁵⁵	çaŋ ⁴⁴ i ⁴⁴	t+E13+a+2	†ae ⁴² li ³⁵	lu55gaw114i33	ła ⁵³ tcoŋ ⁴²	za 龙 华	hu ⁴⁴ Dji ¹¹	hu ⁴⁴ Ban ⁴⁴	tfin ^{s3} de ³⁵	la ²³ çuŋ ²²	ła ^{ss} tçaŋ⁴
八月	+a24zu55	10302U50	Zi ²⁴ †i ⁴⁴	mpa ⁴² ła ⁴²	mpa ⁵⁵ li ³⁵	lu552ñi314i33	†q932i22	zi 龙 定	sεη ³³ πei ²¹	sen33 fa:m33	tsi31 4a24	la ³³ za ⁵³	+a ³⁵ ji ⁴²
九月	ła²⁴tsəu³⁵	ła ³⁵ tsu ⁵⁵	tcua ³³ fi ⁴⁴	nä³¹+a⁴²	nə ¹¹ li ³⁵	lu ⁵⁵ dzha ³⁵ fi ¹¹	†a53t6031	gt 姻 尾	theul nei 42	theu" ta:m35	tliss mut+2	la ³³ tca ⁵⁵	+a55ko33
十月	+a24?la55	+a357la55	kou ²⁴ ti ⁴⁴	ŋĕ ⁵⁵ ła ⁴²	ŋu ⁵³ li ³⁵	lu55ghau314i33	†a53ku22	2a				la ³³ tçu ⁵³	+a55ku42
十一月	ła²⁴juŋ³¹	†a35joŋ31	n ₂ 031+i44	sə ¹³ fa ⁴²	SO ⁴² li ³⁵	lu ⁵⁵ ghau ³¹ ?i ⁵⁵ fi ³³	łasstonss	tu 双龙	tjhau ³³ ni ⁴²	tjhau ³³ San ³³	tes24 nin31 da24	la ³³ ce ³³ zu ²¹	ła ⁵⁵ ku ⁴² je
十二月	+a247nen31	ta ²⁵ ?nen ²¹	tşo554i44	lu13+a42	[a ⁴² li ³⁵	lussghau ²¹ ?ass t i ²²	†a532022	油油岭	heg ⁴⁴ 16i ²²	heg ⁴⁴ hom ⁴⁴	tsjan ⁴⁴ no ⁴²	la ³³ mja ⁵³	ła ^{ss} ku ⁴² ?
今年	nen³¹naŋ⁵⁵	nen ?noŋ55	con44na55	shō42nā13	tcaŋ55na42	cau ³³ na ⁵⁵	tcu ⁵³ nen ⁴⁴	tsurmer	iju-norj	njeu nerj	gu na-	ne ⁵⁵ na ¹³	m³¹tcan⁵
去年	muŋ ⁵⁵ nen ³¹	nen³¹mu³¹?noŋ⁵⁵	7a43tşe43na55	qə ⁰² la ³¹ 5 ¹ 0 ⁴²	tcaŋ ⁵⁵ pze ³²	?a ³³ pu ⁵⁵ na ³³	teu53pa35	ņ³5pa53tsu³3	ŋju⁴⁴?ε³³	njeu44so13	ŋ.u°¹?o¹³	ne ⁵⁵ kua ³³	qa³³tcaŋ¹
明年	p+931nen31	pə ^{to} nen³¹	ə ²⁴ coŋ ⁴⁴	ta555hō42nā13	tcaŋ55sæn22	n-ĥau ¹³ na ⁵⁵	tuu31teu53	səw ⁵³ ku ⁵³ tsu ³³	ŋju⁴⁴qaŋ³³	po ¹³ njeu ⁴⁴	pe ²² ŋ.u ⁴⁴	ne ⁵⁵ qæ̂ ⁴⁴	po³5tcaŋ:
現在	than33nan35	ce ³¹ ?non ⁵⁵	tan ⁵⁵ na ⁵⁵	sha31nã13	sei ²² na ⁴²	nasonios	ma³¹nen⁴⁴	za ¹³ ?nei ⁵⁵	can ³¹ non ³⁵	ha53nen33	sei ³¹ na ⁵³	poi ⁵³ na ²³	m31ke35

Figure 3: Contrasting Chén's original data with the table in Wiktionary

All methods have either been implemented and published before, or are shared along with the slides and the handout for this talk. Since this is work in progress, however, we warn users that both data and code will be in flux for some time, but we will make sure that both data and code can always be readily analyzed with our tools. All code, the data we use, and installation instructions can be found at https://github.com/lingpy/calc-workflow. We ask those interested in testing our methods to use our issue-tracker on GitHub in case they face difficulties of any kind. "" HEAD In this talk, we present the workflow with a subset of 10 varieties of the Hmong-Mien languages in Chén's sample, for which we selected a subset of 313 concepts. The concepts were selected by checking the overlap with the current 504 concept list of the Burmish Etymological Database project (headed by Nathan W. Hill, data online at https://dighl.github.io/burmish). The languages were selected for some general reasons, like lexical coverage, geographic distribution, or basic diversity, but not with the specific "eye" of a historical linguist who would select languages to explore the history of a language family. We would be glad about any additional recommendations, if scholars feel competent to give us advice in this context. The geographic locations are shown in the Figure 4.

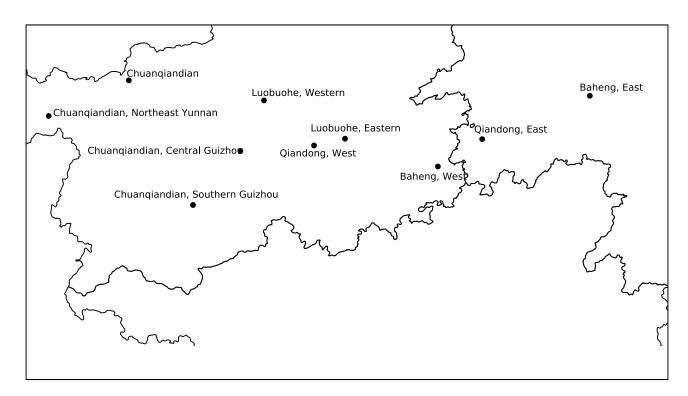


Figure 4: Language geographic locations

3 Illustration of the Workflow

3.1 From Raw Data to Segmented Data

When searching for sound correspondence patterns, we can safely assume that the data is a wordlist; a lexical dataset contains vocabularies which is translated into various languages. The existing wordlists have various presentations, such as, the orientations of data or the usage of seperators of synonyms (3). Due to idiosyncratic formats, linguistic datasets often lack interoperability and are therefore not reusable. Following the *Fair* guiding principles of scientific data management from Wilkinson et al., "Findable, Accessible, Interoperable, and Reusable" (Wilkinson et al. 2016), we convert our raw data into *Lingpy Wordlist* format. The format has the following guidelines:

- · A tab-separated input file.
- First row serves as a header and defines the content of the rest of the rows.
- One value per cell, therefore, synonyms are divided into different rows.
- Four mandatory columns: unique identification numbers for each row, the language name (DOCULECT), the comparison concept (CONCEPT), the original transcription (International phonetic alphabet,

IPA

- , FORM or VALUE).
- TOKENS-columns should supply the supply the transcriptions in space-segmented form.

Many existing tools make use of *Lingpy wordlist* format, including the tools we demonstrate here in this talk.

3.2 From Segmented Data to Cognate Sets

Once the data is segmented and provided in the long table format as it is required by the LingPy software package, as described in our tutorial (List et al. 2018a), we can use LingPy's partial cognate detection method to infer partial cognates in our linguistic data. Partial cognates are hereby understood as cognate assessments *per morpheme* in our data, as opposed to cognate assessments *per word*. While it has always been clear to scholars working in the field of South-East Asian linguistics that cognacy should rather be assigned on the level of the morpheme than on the level of full words, given that the high degree of compounding would easily complicate the identification of cognate relations, automatic methods, and specifically phylogenetic reconstruction approaches usually still assume a rather naive one-word-one-cognate relation (List 2016).

In our framework, we explicitly address this problem by adopting a numerical annotation format in which each morpheme instead of each word form is assigned to a specific cognate set (Hill and List 2017). This framework is illustrated in Figure 5, where we contrast word forms for "yesterday" in five Burmish varieties, indicating their detailed "cognate relations". In the first "traditional" style of cognate coding, we would proceed in a *strict* way, only allowing those words which are completely cognate in all their morphemes to be judged as cognates. In the second, *loose* cognate annotation, we judge all words that are in a *connected component* in our shared morpheme network to be cognate, and in the last column, we show our explicit coding of partial cognacy, in which each morpheme is assigned to one cognate set.

Language	Form	Strict	Loose	Exact
Bola	<mark>a³1</mark> դji³5 ոε?³1	1	1	123
Lashi	a ³¹ ŋjei ⁵⁵ nap ³¹	1	1	123
Rangoon	ma ⁵³ ne ⁵³ ka ⁵³	2	1	030
Xiandao	ņ ³¹ man ³⁵	3	1	3 4
Achang	man ³⁵	4	1	4

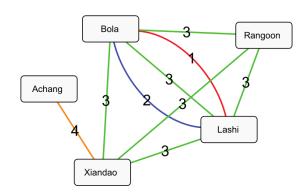


Figure 5: Partial cognacy in Burmish language varieties and different ways of coding (see Hill and List 2017 and further explanations in the main text). coding.

The software package LingPy offers a straightforward algorithm to detect and annotate partial cognates in datasets formatted as long tables. This algorithm by List et al. (2016b) uses techniques for automatic sequence comparison to create a network of similar morphemes for each meaning slot in a given dataset. It then filters those concepts in consecutive stages, with the goal of avoiding that two or more morphemes in the same word for the same language are assigned to the same cluster. In the end, the algorithm outputs the cognate judgments in the same format as indicated above in Figure 5, namely, but assigning each morpheme to a given number, with the number representing that cognate set.

Note that this algorithm works quite well, although it is, of course, not infallible. It reaches between 88 and 90 percent on a test datasets consisting of Bai dialects, Chinese dialects, and dialects of Tujia. With more challenging datasets, the scores will surely drop, but we can expect that the automatic cognate detection is in any case *helpful*, as is easier to correct cognates than to assign them from scratch.

In addition to the cognate detection algorithm, the EDICTOR web-based tool for computer-assisted language comparison (List 2017), freely available at http://edictor.digling.org, can be used to quickly inspect and correct computer-generated cognate sets, by providing a very convenient interface

下巴

Figure 6: Partial cognate annotation within the EDICTOR tool for the word for "chin" in 10 selected Hmong-Mien varieties.

that allows users to quickly assign morphemes to cognate sets. The interface is illustrated in Figure 6.

SUMMARY

- For a realistic annotation of cognate sets, the annotation of partial cognates, by which morphemes are assigned to cognate sets, is the only realistic choice.
- Partial cognates can be automatically identified with help of software, openly available as part of the LingPy software library (lingpy.org, List et al. 2018b) and the algorithm by List et al. (2016b).
- Partial cognates can be annotated consistently with help of the EDICTOR tool (List 2017), online available at http://edictor.digling.org.
- Partial cognates in these frameworks are assigned to morphemes occurring in words with the same meaning, both for algorithmic and for practical reasons.

The frequency of compound words in South-East Asian (SEA) languages. Partial cognacy. New algorithm for cognate detection which does not identify cognate words but instead searches for cognate elements in words.

One morpheme correspondent to one cognate id.

3.3 From Cognate Sets to Alignments

3.4 From Alignments to Cross-Semantic Cognates

As mentioned above in Section 3.2, the partial cognates are only identified for words with the same meaning. This is being done for algorithmic reasons (it would become quite complex to compare all morphemes against each other algorithmically), and for practical reasons, since we believe that it is always better to start from the obvious and save etymologies in historical linguistics, rather than to start from complex ones. Given that semantic shift is a phenomenon for which we dispose of little knowledge with respect to its patterns, we agree explicitly with scholars like Dybo and Starostin (2008) in emphasizing that we should always expect to find clear-cut etymologies within words of the same meaning, even

if we know that more etymologies could be find when searching *cross-semantically*, i.e., among words which differ with respect to their meanings.

There are only a few approaches that try to identify cognates across different concepts, and one could say that the task of *cross-semantic cognate detection* is still one of the open problems in computational historical linguistics. Approaches proposed so far include a rather complex workflow by Wahle (2016), who uses *hidden Markov models* for sequence comparison, and proxies on colexifications, drawn from the database by Dellert and Jäger (2017), to infer cognates across different meaning slots. As this task is not completely evaluated, and only described in a short paper, it is difficult to access its usefulness for our purposes. Another approach is presented by Arnaud et al. (2017), who apply Support Vector Machines trained on form and semantic similarities of word pairs along with a flat clustering algorithm to partition words into cognate sets. While this approach is publicly available and seems to yield promising results, we are not sure to which degree it would help us with our very specific goals of lifting an initially "raw" dataset to a level where we can assess sound correspondence patterns across multiple languages, especially since the algorithms the authors use for cognate detection do *not* take regular sound correspondences into account, and they are also *not* sensitive to partial cognates.

Thus, instead of these previously proposed solutions, we propose our own, rather simple approach to search for cross-semantic partial cognate sets in our data. This approach is based on the well-observed fact that the majority of morphemes in South-East Asian languages with a certain preference for compounding and a high degree of word formation, is highly *promiscuous* (List et al. 2016a: 8f), given that they recur within different words, surfacing in the form of *partial colexifications* (Hill and List 2017: 62). The term *partial colexification* hereby serves as a cover term for morphemes recurring across the lexicon of a language, with no specific distinction being made if they are polysemous or homophonous.

Our search for partial colexifications would not allow us directly to identify cross-semantic cognates consistently, given that sound change may yield different morpheme mergers across different languages. As a result, we cannot take the information from one language alone, but have to smartly summarize all the information on recurring morphemes we can find in our data. The solution for this problem is nevertheless straightforward, and it builds on the idea to not only compare single words, as originally proposed in Hill and List (ibid.), but to compare complete *alignments* instead. As our data is already aligned, and we have identified cognates in a first run, potentially even refined by experts, we can compare whole cognate sets that contain *identical words in the same language*.

If two alignments are completely identical with respect to the words they contain, there is no reason to assign them to different cognate sets, and we can directly assign them to the same cognate class. Even if they are simply homophonous, the assumption of regular sound change will allow us to treat them similarly if we reconstruct the words back to the ancestral language.

The problematic cases are those cases, where we have *incomplete data*. And this is usually rather the rule than the exception. We often will encounter cases where we have two alignments which are only filled in parts with data from the different languages, and we will usually have *missing data* for one or more of the languages in our sample in a given alignment. Thus, when comparing two alignments with each other, we need to make sure that we have at least one word in one language in common.

As an example, consider the data on "son" and "daughter" in five language varieties of our illustration data. As can be seen immediately, two languages show striking *partial colexifications* for the two concepts, Chuanqiandian and East Qiandong. In both cases, one morpheme recurs in the words for the two concepts. In the other cases, we find different words, but if we compare the overall cognacy, we can also see that all five languages share one cognate morpheme for "son" (corresponding to the Proto-Hmong-Mien *tuɛn in Ratliff's reconstruction), and three varieties share one cognate morpheme for "daughter" (corresponding to *mphje^D in Ratliff's 2010 reconstruction), with the morpheme for "son" occurring also in the words for "daughter" in East Qiandong and Chuanqiandian, as mentioned before.

Our workflow for automatically identifying these cases of cognacy is a new algorithm for cross-semantic cognate detection, developed first for the work in the Burmish Etymological Dictionary project

Language	Concept	Form	Cognacy	Cross-Semantic
East Baheng	SON	taŋ ³⁵	1	1
East Baheng	DAUGHTER	p ^h je ⁵³	2	2
West Baheng	SON	$2a^{3}/^{0} + \tan^{35}$	3 1	3 1
West Baheng	DAUGHTER	$ta^{55} + qa^{3/0} + t^{h}jei^{53}$	456	4 5 6
Chuanqiandian	SON	to ⁴³	1	1
Chuanqiandian	DAUGHTER	ntshai ³³	7	7
Chuanqiandian (Central Guizhou)	SON	$t\theta^2/^0 + t\tilde{\theta}^{24}$	8 1	8 1
Chuanqiandian (Central Guizhou)	DAUGHTER	$t\tilde{e}^{24} + {}^{n}p^{h}e^{42}$	9 2	1 2
East Qiandong	SON	tei ²⁴	1	1
East Qiandong	DAUGHTER	$tei^{24} + p^h a^{35}$	9 2	1 2

Table 1: Terms for "son" and "daughter" across five Hmong-Mien varieties.

lead by Nathan W. Hill. In this workflow, we start from all aligned cognate sets in our data, and then systematically compare all alignments with each other. Whenever two alignments are *compatible*, i.e., they have (1) at least one morpheme in one language occurring in both aligned cognate sets, which is (2) identical, and (3) no shared morphemes in two alignments which are *not* identical, we treat them as belonging to one and the same cognate set. We iterate over all alignments in the data algorithmically, merging the alignments into larger sets in a greedy fashion, and re-assign cognate sets in the data.

The results can be easily inspected with help of the EDICTOR tool, for example, by inspecting cognate set distributions in the data. When inspecting the cross-semantic cognates, which we label CROSSIDS in our data, the tool will always show, which cognate sets span more than one concept, and users can directly filter the data and look at the relevant instances. Among the 64 cognate sets reflected in all languages in our sample, we find quite a few cross-semantically recurring morphemes, seven in total (with many more for the whole data). The results are shown in Table 2.

Language	Concept	Form	Morphemes
East Baheng	NOSE	ⁿ pjau ³¹	NOSE
East Baheng	NASAL MUCUS	qa ³ /0 + ⁿ pjau ³¹	qa NOSE
Western Luobuohe	TWO	?ս ³¹	TWO
Western Luobuohe	TWENTY	?u ³¹ + 40 ³¹	TWO zo
Western Baheng	SON	?a ³ / ⁰ + taŋ ³⁵	SON
Western Baheng	SON-IN-LAW	taŋ ³⁵ + wei ³¹	SON wei
Western Baheng	GRANDSON	taŋ ³⁵ + seŋ ³¹	SON seng
Eastern Qiandong	SUN	q ^h aŋ ³³ + nei ²⁴	po SUN
Eastern Qiandong	DAY (NOT NIGHT)	nei ²⁴	SUN
Western Baheng	FAECES (EXCREMENT)	qa ³¹	SHIT
Western Baheng	STOMACH	?a³/º + tɕʰi³⁵ + qa³¹	a tci SHIT
Western Qiandong	ANT	kæ̃ ⁴⁴ + mjɔ ²²	INSECT mjo
Western Qiandong	EARTHWORM	kæ̃⁴⁴+tɕuŋ⁴⁴	INSECT tsung
Eastern Baheng	BIRD	taŋ ³⁵ + nuŋ ³¹	BIRD-A BIRD-B
Eastern Baheng	NEST	۵ ¹¹ + taŋ ³⁵ + nuŋ ³¹	zo BIRD-A BIRD-B

Table 2: Partial cognates among stable concepts with reflexes in all languages in our test datasets. We highlight shared cognates by giving a tentative gloss for them in capital letters in the column *Morphemes*.

SUMMARY

- For a realistic analysis, we need to identify cognates not only within the same meaning slot, but across different concepts, specifically when dealing with languages in which compounding and word formation are very productive.
- We employ a new method that makes use of a comparison of the alignments in readily identified and aligned partial cognate sets to identify those morphemes which recur across different concepts in our data.
- The results can be inspected with help of the EDICTOR, but not directly, by now, only indirectly with help of the browser for cognate sets.
- The interpretation of the results cannot be done automatically, but requires expert assessment with respect to the morphology of the data under consideration.

3.5 From cross-semantic cognates to correspondence patterns

4 Discussion

5 Outlook

References

- Anthony, D. W. and D. Ringe (2015). "The Indo-European homeland from linguistic and Archaeological perspectives". *Annual Review of Linguistics* 1, 199–219.
- Arnaud, A. S., D. Beck, and G. Kondrak (2017). "Identifying cognate sets across dictionaries of related languages". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. (Copenhagen, 09/07–09/11/2017). Association for Computational Linguistics, 2509–2518.
- Barrachina, S. et al. (2008). "Statistical approaches to computer-assisted translation". *Computational Linguistics* 35.1, 3–28.
- Dellert, J. and G. Jäger (2017). *NorthEuraLex (Version 0.9)*. Tübingen: Eberhard-Karls University Tübingen.
- Dybo, A. and G. S. Starostin (2008). "In defense of the comparative method, or the end of the Vovin controversy". In: *Aspekty komparativistiki* [Aspects of comparative linguistics]. Vol. 3: *Aspekty komparativistiki*. Ed. by I. S. Smirnov. Moscow: RGGU, 119–258.
- Hill, N. W. and J.-M. List (2017). "Challenges of annotation and analysis in computer-assisted language comparison: A case study on Burmish languages". *Yearbook of the Poznań Linguistic Meeting* 3.1, 47–76.
- Holm, H. J. (2007). "The new arboretum of Indo-European "trees?. Can new algorithms reveal the phylogeny and even prehistory of Indo-European?" *Journal of Quantitative Linguistics* 14.2-3, 167–214.
- List, J.-M. (2014). Sequence comparison in historical linguistics. Düsseldorf: Düsseldorf University Press.
- (2016). "Beyond cognacy: Historical relations between words and their implication for phylogenetic reconstruction". *Journal of Language Evolution* 1.2, 119–136.
- (2017). "A web-based interactive tool for creating, inspecting, editing, and publishing etymological datasets". In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. System Demonstrations. Valencia: Association for Computational Linguistics, 9–12.

- (2018). Regular cognates: A new term for homology relations in linguistics. Vol. 5. 8.
- (2019). "Automatic inference of sound correspondence patterns across multiple languages". *Computational Linguistics* 1.45, 137–161.
- List, J.-M., J. S. Pathmanathan, P. Lopez, and E. Bapteste (2016a). "Unity and disunity in evolutionary sciences: process-based analogies open common research avenues for biology and linguistics". *Biology Direct* 11.39, 1–17.
- List, J.-M., P. Lopez, and E. Bapteste (2016b). "Using sequence similarity networks to identify partial cognates in multilingual wordlists". In: *Proceedings of the Association of Computational Linguistics* 2016 (Volume 2: Short Papers). Association of Computational Linguistics. Berlin, 599–605.
- List, J.-M., S. Greenhill, C. Anderson, T. Mayer, T. Tresoldi, and R. Forkel, eds. (2018a). *CLICS: Database of Cross-Linguistic Colexifications*. url: http://clics.clld.org/.
- List, J.-M., S. Greenhill, T. Tresoldi, and R. Forkel (2018b). *LingPy. A Python library for quantitative tasks in historical linguistics*. uRL: http://lingpy.org.
- Ratliff, M. (2010). Hmong-Mien language history. Canberra: Pacific Linguistics.
- Wahle, J. (2016). "An approach to cross-concept cognacy identification". In: *Proceedings of the Leiden Workshop on Capturing Phylogenetic Algorithms for Linguistics*. "Capturing Phylogenetic Algorithms for Linguistics" (Leiden, 10/26–10/30/2015). Ed. by C. Bentz, G. Jäger, and I. Yanovich. Tübingen.
- Wilkinson, M. D. et al. (2016). "The FAIR Guiding Principles for scientific data management and stew-ardship". *Scientific Data* 3, 160018.
- 陳其光, C. Q. (2012). *Miàoyáo yǔwén* 妙药语文 [Miao and Yao language]. Ed. by Anonymous. Běijīng: Zhōngyāng Mínzú Dàxué 中央民族大学[Central Institute of Minorities].