BALTIC LINGUISTICS
ISSN 1503-4621
1 (2010), 215–244

The consonant template in synchrony and diachrony

Bernhard Wälchli
University of Bern

This paper investigates the importance of (unique) consonants in identifying cross-linguistic cognates. It is argued that consonants play a crucial role both in diachrony for the identification of cognates and in synchrony for the identification of stemforms. The same algorithm — alignment of the consonant template (ACT) — is applied both in diachrony for identifying cognates and in synchrony for aligning stemforms. It is argued that identifying cognates is essentially the alignment of their consonant templates. Since the alignment of consonant templates is over-generating, ACT must be strongly constrained by semantics. A method is presented to extract cognates directly from parallel texts which is exemplified and evaluated mainly on the basis of Lithuanian and Latvian. For identifying cognates, a three step procedure is applied: (a) finding semantically equivalent forms (SEF), (b) finding equivalent consonants (EC), and (c) alignment of the consonant template (ACT)

Keywords: Baltic languages, historical-comparative method, cognates, statistical natural language processing (NLP), consonants, autosegmental phonology, semantic equivalents, stemforms

1. Introduction

The editors' preface to this first issue of *Baltic Linguistics* states that there is a historical-comparative bias in Baltic Linguistics[1]. I would like to argue here that this diachronic heritage can be an advantage especially if we understand it as an invitation to try out new lines of research. Historical linguistics and typology do not exclude each other. On the contrary, the two are intimately intertwined especially in the area of grammaticalization studies, and typology has contributed a great deal to overcome the strict separation of synchronic and diachronic ap-

[1] I would like to thank Thomas Mayer (Konstanz), Michael Cysouw (Munich) and Raphael Berthele (Fribourg) for many useful comments. While writing this paper I have been funded by the Swiss National Science Foundation (PP001-114840).

proaches. This paper takes a slightly different approach. It shows that elementary tools in statistical natural language processing (NLP) can be applied with great profit to the diachronic comparison of closely related languages such as Latvian and Lithuanian. The aim is not to reinvent comparative linguistics, rather the objective is methodological. Historical linguistics and typology have in common that much of their methodology is implicit and intuitive. Computers completely lack intuition and must be told every step explicitly. In trying to let computers replicate the job of historical linguists and typologists we can learn more about the foundations of these disciplines, which are so self-evident to historical linguists and typologists that they hardly ever discuss them. Some of the foundations of simple comparative tasks, such as identifying cognates — the topic to be addressed here — can be easily implemented fully automatically, other ones are more difficult to implement. Computational approaches can sharpen our awareness of which aspects of the process are easy and which ones are the really difficult ones. Computational approaches also sharpen the procedural aspects of the endeavor (which first step must have been performed so that another second step can take its output as input).

Comparative linguists hardly ever make fully explicit what they mean by the comparative method. Especially the initial basic steps remain implicit in the description. Textbook exercises are made such that the initial steps have already been done so that students of comparative linguistics hardly ever have to do the first steps themselves. This can be nicely illustrated with Campbell's (2004) popular general introduction to historical linguistics, which in chapter 5 introduces the comparative method in form of a seven-step procedure. The first step is 'Assemble cognates' (we need not be concerned here about the other steps since we will not go any further in this paper) and this step is described very implicitly: "To begin to apply the comparative method, we look for potential cognates among related languages (or among languages for which there is reason to suspect relatedness) and list them in some orderly arrangement (in rows or columns). In Table 5.1, this step has already been done for you..." (Campbell 2004, 126).

This paper deals with three easily implementable aspects in the process of identifying cognates: (i) the identification of cross-linguistic functional equivalents, (ii) the identification of most elementary sound correspondences, and (iii) the alignment of consonant templates as evidence for diachronic relationship. Historical linguists traditionally emphasize that formal regularities (such as sound laws) are much more relevant than functional regularities. However, it is argued here that the basis for any kind of cross-linguistic comparison, be it in comparative linguistics or in typology, is functional equivalence. In order to discover formal regularities, such as sound laws, it must first be assured that the forms to be compared are semantically closely related. This same functional requirement as a basis for establishing connections between forms holds in a very similar vein also for identifying pairs of stems in language acquisition. Evidence for this parallelism between synchrony and diachrony is supplied by applying exactly the same algorithm to cross-linguistic (diachrony) and language-specific data (synchrony). The basic hypothesis argued for is that finding cognates in closely related languages and identifying stemforms related by ablaut or umlaut is very much the same kind of task. Both rely on consonant templates and both are heavily dependent on constraining the input semantically, otherwise they would strongly over-generate.

This article describes an algorithm for the ALIGNMENT OF CONSONANT TEMPLATES (ACT), which has been originally developed for identifying stemforms of a lexeme with internal inflection (such as English *man men*, *stand stood*, or Lithuanian *krinta* 'fall:PRS3' *krito* 'fall:PST3'), but is applied here to semantic equivalent forms in related languages in order to identify diachronic cognates. ACT is combined with two other algorithms: (i) finding semantically equivalent forms in parallel texts (SEF), and finding equivalent consonants (EC). This allows us to run ACT automatically on parallel texts without any previous manual analysis. This automatic identification of cognates is illustrated mainly on the basis of Latvian and Lithuanian, but also some other pairs of closely related languages.

I would like to emphasize right from the beginning that the performance of this automatic procedure is much weaker than what comparative linguists achieve by manual labor and it is not the aim of this article to compete with the traditional historical-comparative method. Rather, I will use ACT to make some general claims. The principle underlying ACT is that related forms share the same consonant template and are related by the alignment of unique consonants while correspondences in vowels are of secondary importance. In identifying pairs of forms, however, ACT is highly over-generating. It aligns *man* and *moon* as easily as *man* and *men*. This is where semantics comes into play. Both in diachrony and in synchrony, ACT must be heavily constrained by semantics. Only if we make sure that the forms that are compared are very similar in their lexical meaning does it make sense to compare them, be it in order to identify cognates in diachrony or stemforms of the same lexeme in synchrony. Put differently, what (competent) speakers are able to do and what (competent) historical linguists are able to do is very similar.

The basic assumptions are repeated below:

(i) For identifying cross-linguistic cognates consonants are more important than vowels.

(ii) Consonants are more important than vowels for identifying stem-forms exhibiting internal inflection, such as English *find* (present) *found* (past), Lithuanian *kelia* 'rises' *kėlė* 'rose'.

(iii) The mechanism underlying (i) and (ii) is virtually the same: the identification of unique consonants in fixed order.

(iv) Since the mechanism in (iii) is over-generating, it must be heavily constrained by semantics.

2. The algorithm

The algorithm described in this section can extract good candidates for cognates in language pairs fully automatically from parallel texts. The method works only for closely related languages where cognates are easy to identify. The parallel text used is the New Testament (NT), which has two crucial advantages: (i) it is freely available electronically in a large number of languages and (ii) it is aligned on the level of verses which means that we can skip a complex processing step: the sentence-to-sentence alignment in parallel texts. The algorithm consists of three subsequent parts:

(a) Finding semantically equivalent forms (SEF)
(b) Finding equivalent consonants (EC)
(c) Alignment of the consonant template (ACT)

This is illustrated with two simple examples. First, we identify semantically equivalent forms in parallel texts by SEF; for instance, Latvian *sacīdams* and Lithuanian *sakydamas* 'saying' (simultaneous converb for same subject masculine singular) or Latvian *gara* and Lithuanian *dvasios* 'spirit (GEN:SG)'. Next we find out how Latvian consonant characters correspond to Lithuanian consonant characters (EC). EC tells us that Latvian <c> does not correspond to Lithuanian <c>, but rather to Lithuanian <k>. Finally, we try to align the forms by ACT which is successful in the pair *sakīdams sakydamas* (recall that we first have to replace Latvian <c> by <k>) but not in the pair *gara dvasios* which is why the former pair is likely to be cognate while the latter is not.

The main reasons why the algorithm only works in closely related languages and only to a certain extent are the following:

- Orthography is no ideal input. The performance would be better with phonological input. Especially it is bad if di- or trigraphs are used for one consonant phoneme, such as Latvian <dž> or German <sch> (for a much more sophisticated approach see Cysouw & Jung 2007).
- The part (b) EC is rather primitive. It assumes that any consonant in language A exactly corresponds to one consonant in language B in all contexts. As is well known, this does not hold true; sound laws are highly dependent on phonological environment.
- The algorithm compares any form with any form, not making any distinction between core vocabulary (which is more likely to be inherited) from culturally dependent terms (often loanwords) or proper names (mostly rather useless for diachronic comparison). Only in closely related languages, cognate forms are dominant over loanwords and proper names.
- In morphologically complex languages, it can happen that many forms of the same lexemes are extracted. These can cause accidental consonant correspondences. If we have many pairs from the same non-cognate lexemes, such as *viņš jis* 'he:NOM', *viņa jo* 'he:GEN', *viņam jam* 'he:DAT', *viņu jį* 'he:ACC', it may happen that Latvian <ņ> is equated wrongly with Lithuanian <j>.

Put differently, in our automatic approach we neglect some basic and ancient principles of the comparative method for reasons of convenience (considering them would mean that the algorithm could not be run automatically). These neglected principles are the following:

(i) Use phonology rather than orthography.

(ii) Establish the exact phonological conditions of sound correspondences.
(iii) Disregard all forms that are likely to consist of more recent cultural layers.
(iv) Compare lexemes (stems or roots) rather than wordforms; compare grammatical and lexical components of words separately.

Even though these principles are well established and largely undisputed, it may be useful to see how much harm is done if they are neglected. It is shown here that the performance of a cognate identifying algorithm in Latvian and Lithuanian can be pretty high even if these basic principles are disregarded. The three sub-algorithms, ACT, EC, and SEF, will now be discussed in inverted order.

2.1. Alignment of the consonant template (ACT)

Let us assume we have already a list of semantically equivalent Lithuanian and Latvian forms (this is done by SEF; some examples are given in Table 1 columns 1 and 2) and we have established rough correspondences between consonant characters (this is done by EC; Table 1, column 3: for instance, Latvian *ļ* must be compared with Lithuanian *l* and Latvian *c* with Lithuanian *k*). This is the input we need for ACT to be described here (Table 1 column 4). The only column added manually in Table 1 is the gloss.

ACT extracts the consonants and tries to align them on the basis of identical consonants. If ACT is successful it adds the vowels and the remaining consonants to the template so that variants end up in brackets with the notation [x|y] where *x* is the sequence of Form 1 and *y* the sequence of Form 2. Thus, [x|y] is read as “x or y”; *sak[au|u]* is read “sakau or saku”. If ACT is not successful, it returns an empty string. In the examples given in Table 1, ACT is successful except in two pairs of forms which are no cognates (the words for ‘son’ with two different inherited etyma and the words for ‘human being’ where the Latvian word has been borrowed from East Slavic). Of course, ACT can by no means replace the comparative linguist; it over-generates candidates for cognates if applied to domains where it does not make sense to compare forms diachronically. ACT aligns *Paulius* and *Pāvils* even though “Paul” has hardly ever been a popular Proto-Baltic name.

Table 1: ACT with some Latvian and Lithuanian forms

Lithuanian	Latvian	Latvian with 'Lithuanized' consonants	Alignment	Gloss (of both forms)
dievas	dievs	dievs	diev[a|]s	God:NOM:SG
broliai	brāļi	brāli	br[o|ā]l[iai|i]	brother:NOM:PL
sakau	saku	saku	sak[au|u]	say:PRS1SG
mūsų	mūsu	mūsu	mūs[u|ų]	we:GEN
jums	jums	jums	jums	you[PL]:DAT
paulius	pāvils	pāvils	p[au|āvi]l[iu|]s	Paul:NOM
sūnus	dēls	dēls		son:NOM:SG
tiesų	patiesi	patiesi	[pa|]ties[i|ų]	verily
žmogaus	cilvēka	kilvēka		human.being:GEN:SG
kiek	cik	kik	k[ie|i]k	how many
kitą	citu	kitu	kit[u|ą]	other.ACC.SG

ACT has originally been designed by me for another purpose: for an automatic identification of internal inflection in pairs of stemforms with identical lexical meaning, such as the English nominal singular plural pairs with umlaut, such as *m[a|e]n, br[e|o]th[|e]r[en|]*, the English present and past forms in strong verbs (Table 2) and the interdigitation in Semitic non-concatenative morphology, such as in Maltese broken plurals (Table 3).

Table 2: English strong verbs aligned with ACT

English present	English past	Alignment
drink	drank	dr[a|i]nk
fall	fell	f[a|e]ll
feed	fed	f[e|ee]d
find	found	f[i|ou]nd

Continuation of Table 2

<table>
<tr><th>English present</th><th>English past</th><th>Alignment</th></tr>
<tr><td>know</td><td>knew</td><td>kn[e|o]w</td></tr>
<tr><td>run</td><td>ran</td><td>r[a|u]n</td></tr>
<tr><td>sit</td><td>sat</td><td>s[a|i]t</td></tr>
<tr><td>take</td><td>took</td><td>t[a|oo]k[e|]</td></tr>
<tr><td>write</td><td>wrote</td><td>wr[i|o]t[|e]</td></tr>
</table>

Table 3: Maltese broken plurals aligned with ACT

<table>
<tr><th>Singular</th><th>Plural</th><th>Alignment</th><th>Meaning</th></tr>
<tr><td>abjad</td><td>bojod</td><td>[a|]b[|o]j[a|o]d</td><td>‘white’</td></tr>
<tr><td>belt</td><td>bliet</td><td>b[e|]l[|ie]t</td><td>‘city’</td></tr>
<tr><td>ġisem</td><td>iġsma</td><td>[i|]ġ[|i]s[|e]m[a|]</td><td>‘body’</td></tr>
<tr><td>kelma</td><td>kliem</td><td>k[e|]l[|ie]m[a|]</td><td>‘word’</td></tr>
<tr><td>ktieb</td><td>kotba</td><td>k[o|]t[|ie]b[a|]</td><td>‘book’</td></tr>
<tr><td>sena</td><td>snin</td><td>s[e|ni]n[a|]</td><td>‘year’</td></tr>
<tr><td>sultan</td><td>slaten</td><td>s[|u]l[a|]t[e|a]n</td><td>‘king’</td></tr>
<tr><td>tabib</td><td>tobba</td><td>t[a|o]b[i|]b[|a]</td><td>‘doctor’</td></tr>
<tr><td>tarbija</td><td>trabi</td><td>t[a|]r[|a]b[ija|i]</td><td>‘baby’</td></tr>
<tr><td>xahar</td><td>xhur</td><td>x[a|]h[a|u]r</td><td>‘month’</td></tr>
</table>

For the alignment of Baltic present and past forms, ACT is only partly suited since there are many cases where there are other processes involved than ablaut (present stem extensions *-st, -n(-)*, metathesis). Table 4 lists some examples from Lithuanian.

Here follows a brief description of how ACT works in detail. It can only align pairs of forms, not — at least not directly — groups of many forms (this would have to be done iteratively by aligning pairs). It takes two arguments, the two forms to be aligned, which are ordered alphabetically by convention (for instance *run ran* > *ran run*).

Table 4: Lithuanian first conjugation present and past forms (only partly successfully) aligned with ACT

Present 3rd person	Past 3rd person	Alignment	Meaning
eina	ėjo		‘goes’
ima	ėmė	[i|ė]m[a|ė]	‘takes’
junta	juto	j[un|u]t[a|o]	‘feels’
kelia	kėlė	k[e|ė]l[ia|ė]	‘raises’
klysta	klydo	kly[do|sta]	‘strays’
lyja	lijo	l[i|y]j[o|a]	‘rains’
skrenda	skrido	skr[en|i]d[a|o]	‘flies’
temsta	temo		‘darkens’
trokšta	troško		‘is thirsty’
vagia	vogė	v[a|o]g[ia|ė]	‘steals’

Because not all forms contain only unique consonants and since ACT relies on the idea of aligning unique consonants, the next step is a trick: a hidden dissimilation process, computationally easiest to implement with upper case letters (all input forms are lower case). Thus, the forms *brethren brother* are turned into *bRethren bRother* or *brethRen brotheR*. For the alignment of stemforms, for which ACT has been designed, it is in practice sufficient with very infrequent exceptions to account for two identical consonants per form. This is because stems tend to avoid identical consonants except in gemination and reduplication and related processes (the similar place of articulation avoidance principle, claimed to be universal by Pozdniakov & Segerer 2007, see Mayer *et al.* 2010 for a confirmation in a huge world-wide typological sample)[2]. In many cases it does not matter whether the dissimilation is applied forward or backward. The pairs *bRethren bRother*

[2] For readers familiar with comparative linguistics, the dissimilation process implemented here is similar to Grassmann’s Law (Graßmann 1863) in Greek and Indic according to which an aspirated consonant followed by an other aspirated consonant in the next syllable loses its aspiration (Greek *$p^{h}ep^{h}uka$ > $pep^{h}uka$ ‘I have grown’). The difference is that any second occurrence of a consonant is dissimilated irrespective of the syllable structure.

and *brethRen brotheR* can be equally well aligned, similarly *faLl feLl* and *falL felL*. However, things get more complicated as soon as the non-unique consonant does not occur with the same number of occurrence in both forms to be paired. This is the case, for instance, in Maltese *sena snin* 'year', which can be aligned to *s[e|ni]n[a|]* or *s[e|] n[a|in]*. The reason for consonant insertion is the following. Maltese broken plurals can be classified according to their CV structure and *snin* 'years' follows the pattern $C_1C_2VC_3$. Because the root provides only two consonants, one consonant must be doubled to match the $C_1C_2VC_3$-pattern (see Schembri 2006). We will not further consider the question here which of the two alignments *s[e|ni]n[a|]* or *s[e|]n[a|in]* is correct or whether they are both equally good. All that matters here is that the forms can be aligned.

The next step is to decompose the forms into CV-structure, consonants, and vowels. Thus, *bRethren bRother* is turned into *CCVCCCVC bRthrn ee CCVCCVC bRthr oe*. For this step ACT must know which characters are consonants and which vowels. This language-particular information can either be specified manually or it can be determined automatically for instance with Sukhotin's algorithm (Sukhotin 1962; Xanthos 2007)[3].

In a next step we count the number of matching and non-matching consonants irrespective of their order. In *brethren brother* there are five matching (b,h,r,r,t) and one non-matching (n) consonant. If there are less matching consonant tokens than either of the two forms has non-matching consonants, alignment is not further attempted and an empty string is returned.

Next we have to check whether the consonants occur in the same order. The consonants are given numbers according to their occurrence in the first form, in the second form and vice versa, and the order must always be monotonously ascending. If this is not the case, alignment

[3] Actually Sukhotin's algorithm works only with phonological input. Lithuanian orthography is problematic, because <i> is used both as a vowel and to mark palatalization of consonants. Latvian orthography is better in this respect, but Sukhotin's algorithm sometimes has problems because /s/ is very frequent and because /a/ is clearly more frequent than other vowels. Better results can be reached with other methods, but it would be overkill for this paper to describe them in detail, in particular because it is easy to tell the computer which characters should be treated as vowels and consonants.

is not further attempted, an empty string is returned. Put differently, ACT cannot account for metathesis.

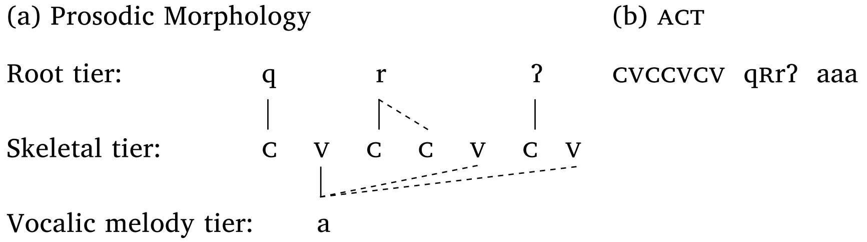
If it has been assured that the two forms can be aligned, what remains to be done is to insert the remaining vowels and the consonants that have been left over. We get thus *bR[e|o]th[|e]r[en|]*. After removing the dissimilation (*br[e|o]th[|e]r[en|]*) we can remove unnecessary brackets (for instance, *j[u|u]ms > jums* Lithuanian/Latvian 'you[PL]:DAT') wherever x and y in [x|y] are identical. Table 5 exemplifies ACT in all subsequent steps with two examples from Table 1.

Table 5: Two Lithuanian Latvian forms to be aligned with ACT

Input	rankas rokas	sūnus dēls
Alphabetic order	rankas rokas	dēls sūnus
Dissimilation	rankas rokas	dēls Sūnus / dēls sūnuS
"Autosegmentation"	CVCCVC rnks aa CVCVC rks oa	CVCC dls ē CVCVC Sns ūu
More matching consonants	rks 3 matching n 1 non-matching	s 1 matching; more non matching cons.: dl 2 Sn 2 -> EXIT
Monotonously ascending order	r>1 k>2 s>3 in both forms	
Alignment	r[an|o]k[a|a]s	
Undo dissimilation	r[an|o]k[a|a]s	
Debracket	r[an|o]kas	

The term "autosegmentation" in Table 5 suggests that ACT rests on the idea of autosegmental or prosodic morphology. This is not quite the case. There are some major differences. Prosodic morphology (McCarthy 1979, Katamba 1993, 165) developed from autosegmental phonology (Goldsmith 1976) and assumes that there are generally three different tiers in roots, a root tier, a skeletal tier and a vocalic melody tier, as exemplified in Figure 1a for the Arabic verb form *qarraʔa* 'he caused to read'. Tone languages have additional layers for tone (autosegmental phonology has originally been developed to account for tone). It is argued for Semitic that the meaning of verbal lexemes is signaled at the root tier, the skeleton tier provides a canonical shape associated with a particular grammatical meaning (such as causative), and that the vocalic melody tier provides inflectional and derivational information. Phonemes can spread to unassociated C and V slots (dotted lines) according to particular rules. ACT (Figure 1b) is more primitive, there is no spreading.

Figure 1: Prosodic morphology vs. ACT



In prosodic morphology, prefixes and suffixes are added by means of additional tiers containing both consonant and vowel phonemes and the tiers are then cyclically conflated until only the skeletal and a conflated consonant-vowel tier remain. The major difference to ACT is that prosodic morphology is used to generate single forms, but ACT to analyze — and not to generate — pairs of forms. In ACT autosegmentation is just an intermediate step to identify a basis for alignment. Put differently, in ACT, the skeleton is the consonants rather than the CV-sequence. The CV sequence is only used to remember in which order consonants and vowels must be recombined after successful alignment.

2.2 Finding equivalent consonants (EC)

In searching for stemforms of a paradigm in one language it can be assured that forms to be aligned share the same phonological system. In cross-linguistic comparison, however, it is much less clear what identical consonants are. Since I am using orthography here, I have no information about phonetic similarity of consonants. What is used is the distributional similarity of consonant graphemes. Before applying ACT any consonant in language A is replaced by the consonant in language B that matches its distribution best. Let us illustrate this with an example:

In 1’000 Lithuanian-Latvian wordform equivalent pairs we look for the best equivalent for Lithuanian <t>. There are 206 forms where both Latvian and Lithuanian contain a <t>, 315 where Lithuanian forms contain at least one <t>, and 279 where Latvian forms have a <t>. There are a number of collocation measures that can be used to measure the degree of fit, such as Jaccard (Dice), log-Likelihood, and T-score (see Manning & Schütze 1999, chapter 5 for a survey). Here we use T-score (see Dahl 2007 for an application in typology), but other collocation measures yield similar performances.

(1) $$T = \frac{prob(A,B) - prob(A) \times prob(B)}{\sqrt{\frac{1}{n} \times prob(A,B)}} = \frac{\frac{a}{n} - \frac{x}{n} \times \frac{y}{n}}{\sqrt{\frac{1}{n} \times \frac{a}{n}}} = \frac{\frac{206}{1000} - \frac{315}{1000} \times \frac{279}{1000}}{\sqrt{\frac{1}{1000} \times \frac{206}{1000}}} = 8.229$$

For our example, T is 8.229 which is a considerably higher value than for Lithuanian <t> and Latvian <p> (1.505) which would be the next best match for Lithuanian <t>. The best corresponding consonant characters are determined in both directions. (2) are the Lithuanized-Latvian characters corresponding to Latvian characters, (3) the Lithuanian to Lettonized-Lithuanian correspondence pairs.

(2) Latvian to Lithuanized-Latvian[4]
ģ g, ķ l, ļ l, ņ j, š č, c k, b b, d d, g š, f f, k k, j j, m m, l l, n n, p p, s s, r r, t t, v v, z z, ž m,

(3) Lithuanian to Lettonized-Lithuanian
š g, b b, d d, g ģ, f f, m m, k k, j ņ, č t, l ģ, n n, p p, s s, r r, t t, v v, z z, ž v,

EC is particularly useful if the two languages to be compared have different writing systems, as, for instance, in the pair Russian Croatian.

[4] Obvious errors are underlined. They are mostly due to rare consonants where distribution is no good cue. Latvian <ķ>, for instance, only occurs in five forms, all of them loanwords.

(4) and (5) give the correspondent characters and Table 6 lists some results of the algorithm for Russian and Croatian

(4) Russian to Croatianized-Russian
с s, р r, т t, х h, ф f, ч č, ц c, щ v, ш š, б b, г g, в v, д d, з z, ж ž, й j, л l, к k, н n, м m, п p,

(5) Croatian to Russianized-Croatian
ć б, č ч, đ л, š ш, c ц, b б, d д, g г, f ф, h х, k к, j в, m м, l л, n н, p п, s с, r р, t т, v в, z з, ž ж,

503 of 1000 Russian forms can be aligned with Croatian forms extracted by SEF (see 2.3. below). Aligned forms are almost exclusively cognates and parallel loans (including proper names). Non-cognate forms and many forms with non-cognate derivations (such as *женщина žena* 'woman') are not aligned.

If we, however, try to align two unrelated languages, such as Finnish with Lithuanian, as expected, the result is not particularly promising. There are only 53 of 1'000 aligned forms, 31 of which are names. That the consonant correspondences are partly correct (6–7) is entirely due to proper names.

If we start comparing more distantly related languages, the result of EC is not quite good any more. Still relatively rewarding is the attempt to compare Lithuanian with Russian, which may be due in part also to Eastern Slavic loanwords in Lithuanian. Table 7 lists some examples of correctly identified cognates.

(6) Lithuanian to Russianized-Lithuanian
š с , c с , b б , d д , g г , f ф , h - , k к , j ф , č д , l л , m м , p п , s с , n н , t т , v в , r р , z с , ž з ,

(7) Russian to Lithuanized-Russian
с š , р r , т t , х d , ф z , ч g , ц l , щ t , ш j , б b , г g , в v , д d , з ž , ж v , й k , л l , к k , н n , м m , п p ,

In comparing Lithuanian with German, the performance is lower, but there are still some correct cognates (Table 8):

Table 6: Examples for automatically extracted Croatian-Russian cognates

Croatian	Russian	Russianized Croatian	Alignment	Gloss (of both forms)
čas	час	час	ч[а|а]с	‘hour’
djelo	дело	двело	д[ве|е]л[о|о]	‘thing’
dobro	хорошо	добро		‘good’
drugi	другой	дрuгі	др[u|у]г[і|ой]	‘second, other’
imenom	именем	іменом	[і|и]м[е|е]н[о|е]м	‘name.INS’
iziđe	вышел	ізіле		‘went out’
ljudi	люди	лвuді	л[вu|ю]д[і|и]	‘people’
mjesto	место	мвесто	м[ве|е]ст[о|о]	‘place’
svjedočanstvo	свидетельство	сввedочанство		‘testimony’
učitelju	учитель	uчітелвu	[u|у]ч[і|и]т[е|е]л[вu|ь]	‘teacher’
uvijek	всегда	uвівек		‘always’
vaš	ваш	ваш	в[а|а]ш	‘your[PL].M.SG.NOM’
vjera	вера	ввера	[в|]в[е|е]р[а|а]	‘faith’
žena	жена	жена	ж[е|е]н[а|а]	‘wife’
žena	женщина	жена		‘woman’

Table 7: Examples for automatically extracted Lithuanian-Russian cognates

Russian	Lithuanian	Lithuanized Russian	Alignment	Meaning
брата	brolio	brata	br[olio|ata]	‘brother:GEN’
будьте	būkite	bydьte	b[ūki|ydь]t[e|e]	‘be:IMP2PL’
день	diena	denь	d[ie|e]n[a|ь]	‘day’
земля	žemė	žemlя	ž[e|e]m[ė|lя]	‘earth’
знаю	žinau	žnaю	ž[i|]n[au|аю]	‘know:PRS1SG’
камнями	akmenimis	kamnяmи	[a|]k[|a]m[e|]n[i|я]m[is|и]	‘stone:INS:PL’
когда	kada	kogda	k[a|og]d[a|a]	‘when’
крови	kraujo	krovи	kr[aujo|ovи]	‘blood:GEN’
никому	niekam	nиkomy	n[ie|и]k[a|o]m[|y]	‘nobody:DAT’
свет	šviesa	švet	šv[iesa|et]	‘light’
сердца	širdis	šerdla	š[i|e]rd[is|la]	‘heart:GEN’
славу	šlovę	šlavy	šl[o|a]v[ę|y]	‘glory:ACC’
смерть	mirtis	šmertь	[|š]m[i|e]rt[is|ь]	‘death’
тому	tam	tomy	t[a|o]m[|y]	‘he:DAT’
три	tris	trи	tr[is|и]	‘three’
умер	mirė	ymer	[|y]m[i|e]r[ė|]	‘died:PST’

Table 8: Examples for automatically extracted Lithuanian-German cognates

<table>
<tr><th>Lithuanian</th><th>German</th><th>Germanized-Lithuanian</th><th>Alignment</th><th>Meaning</th></tr>
<tr><td>mano</td><td>mein</td><td>mano</td><td>m[a|ei]n[o|]</td><td>‘my’</td></tr>
<tr><td>naktį</td><td>nacht</td><td>nactį</td><td>nac[h|]t[|į]</td><td>‘night(ACC)’</td></tr>
<tr><td>pasiuntė</td><td>sandte</td><td>pasiuntė</td><td>[pa|]s[iu|a]n[|d]t[ė|e]</td><td>‘send:PST’</td></tr>
<tr><td>pastatė</td><td>stellte</td><td>pastatė</td><td>[pa|]st[a|ell]t[ė|e]</td><td>‘put. upright:PST’</td></tr>
<tr><td>širdį</td><td>herz</td><td>hirtį</td><td>h[e|i]r[z|tį]</td><td>‘heart(ACC)’</td></tr>
<tr><td>sūnaus</td><td>sohnes</td><td>sūnaus</td><td>s[oh|ū]n[e|au]s</td><td>‘son:GEN’</td></tr>
</table>

When comparing two other distantly related languages from another language family, Finnish and Hungarian, EC detects at least the characteristic correspondence of Finnish *k* with Hungarian *h*, for instance Finnish *kolme*, Hungarian *három* ‘three’ > *károm*, aligned to *k[ol|áro]m[e|]*.

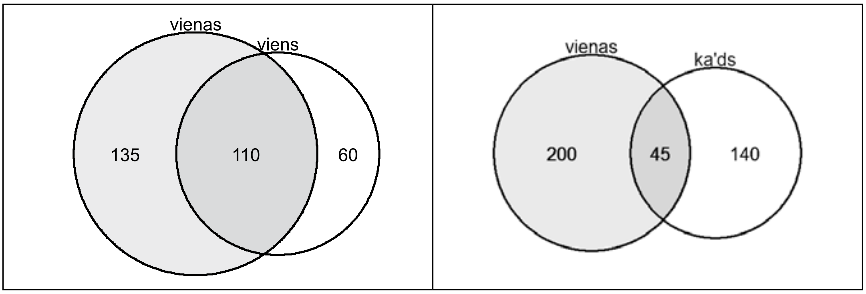
However, it cannot be denied that EC is clearly the weakest chain link in the algorithm. Especially if run with orthographic rather than phonological input, it cannot be expected that all sound correspondences are made correctly.

Campbell (2004, 127) argues that establishing sound correspondences is the second step in the comparative method which applies only after cognates have been identified. It is argued here that it is impossible to identify cognates without establishing any kind of sound correspondences. Of course, the two steps depend on each other to a certain extent, but identifying at least some sound correspondences correctly is a precondition for identifying cognates.

2.3 Finding semantically equivalent forms (SEF)

While EC in 2.2 identifies consonant graphemes on the basis of their distribution across wordforms, SEF — to be discussed here — does exactly the same thing by comparing the distribution of wordforms across verses. (The term 'verse' is used because the text it is applied to is the NT.) The same collocation measure, T-score, is used here again. Figure 2 illustrates how the best Latvian equivalent for Lithuanian *vienas* 'one:NOM:SG:M' is identified, which is Latvian *viens* 'one:NOM:SG:M'. The second best equivalent would be Latvian *kāds* 'some :NOM:SG:M'.

Figure 2: Lithuanian vienas with its best and second best Latvian corresponding form



Lithuanian *vienas* occurs in 245 verses of the NT translation, Latvian *viens* in 170 verses, the intersection is 110 verses. (8) shows how the T-value is calculated. It can already be seen from the Venn diagrams in Figure 2 that *vienas* and *viens* have a more substantial intersection than *viens* and *kāds*.

(8) $$T = \frac{\frac{110}{7959} - \frac{245}{7959} \times \frac{170}{7959}}{\sqrt{\frac{1}{7959} \times \frac{110}{7959}}} = 9.989$$

Finding functional equivalents with SEF is the easier the better the two languages to be compared match in inner form. Put differently, if the languages to be compared have largely the same grammatical categories used in the same way and the same kind of polysemy patterns in their lexemes and if their morphological typology is similar (especially the degree of synthesis), then it is easy to find good functional equivalent forms. It happens to be the case that genealogically closely related languages are usually more similar in inner form than unrelated languages (except in cases of strong areal contacts). What is important to note here is that SEF has a much broader range of application than ACT. Functionally equivalent forms can be extracted from parallel texts in any pair of languages. Table 9 gives some examples for Latvian forms with their best functional equivalents in Lithuanian, Estonian and Early Modern English. While the historical-comparative method can be easily applied to the pair Latvian Lithuanian, it does not make sense for the pair Latvian Estonian even though the functional equivalence of the Latvian-Estonian equivalents is often nearly as good or even better than for Latvian-Lithuanian.

Table 9: Expression form and genealogic relationship

Latvian	Lithuanian	Estonian	Early Modern English
atbildēja	atsakė	vastas	answered
bet	bet	aga	but
bija	buvo	oli	was
cilvēka	žmogaus	inimese	man
daudz	daug	palju	many
dēls	sūnus	poeg	son
dieva	dievo	jumala	god
es	aš	ma	I
ir	yra	on	is
jums	jums	teile	you
jūs	jūs	te	ye
mums	mums	meile	us
nāca	atėjo	tuli	came
redzēja	pamatė	nägi	saw
rokas	rankas	käed	hands
savas	savo	oma	own
tu	tu	sa	thou

The wider applicability of functional equivalence in comparison with formal equivalence makes that the former is much more interesting for large-scale cross-linguistic typological comparison (see Wälchli forthc. for a practical application to measure the similarity of languages in inner form). However, finding functional equivalents is also a precondition for the cross-linguistic comparison of form. It is the basis for historical-comparative investigations in the same vein as it is the basis for functional typology.

3. Evaluating the performance for Latvian and Lithuanian

In this section I will evaluate the results of the automatic cognate identifying algorithm in parallel texts described above for the thousand most frequent Lithuanian forms in the NT with their Latvian equivalents.

For every pair of forms, four attempts are made to apply ACT. The two binary parameters of variation are (a) the hidden dissimilation of identical consonants (whether the first or last consonant is dissimilated, see 2.1 above) and (b) the adaption of consonant characters (whether Latvian consonants are Lithuanized or Lithuanian consonants Lettonized, see 2.2 above). In a clear majority of cases, all four attempts are equally successful or equally unsuccessful. Of 408 aligned pairs, 330 (80.9%) are aligned by all four different methods. Differences arise in case of partial sound shifts. That Latvian <c> corresponds to Lithuanian <k> is recognized only with Lithuanized Latvian consonants, because Latvian /ts/ developed from /k/ before front vowels. Thus, virtually any Latvian <c> corresponds to Lithuanian <k>, but Lithuanian <k> often corresponds to Latvian <k>. Whether backward or forward dissimilation is more successful depends mainly on whether the additional confusing consonant is in a suffix or in a prefix. Lithuanian *manimi* ‘I:INST’ can be aligned with Latvian *mani* ‘I:ACC’ only if the hidden dissimilation is progressive (*maniMi*, not *Manimi*). In evaluating ACT below I only consider whether any of the four attempts in alignment by ACT is successful, not how many of them.

The first step in the evaluation is to consider (by manually checking all pairs) whether SEF finds ‘correct’ functional equivalents. Table

10 shows the results. Of 1000 forms, 118 do not extract a ‘correct’ functional equivalent, and hence it is expected that there should not be any alignment with ACT.

Wrong equivalents occur especially in case of strong collocations. For instance, Lithuanian *karalystės* ‘kingdom:GEN:SG’ is identified with Latvian *dieva* ‘God:GEN’ because “kingdom of God” is a frequent collocation in the NT and Lithuanian *šventoji* ‘holy:NOM:SG:F:DEF’ is identified with Latvian *gars* ‘spirit’ because “holy spirit” is another frequent collocation in the NT. Remember that T-score is a collocation measure. Cross-linguistic distributional equivalents are nothing else than cross-linguistic collocations in parallel texts. Another related source of ‘errors’ are Lithuanian forms inflected for person being identified with Latvian personal pronouns, such as *kalbu* ‘speak:PRS:1SG’ with *es* ‘I’. Similarly some Lithuanian instrumental forms go for the Latvian preposition *ar* ‘with’. This is not unexpected, since there is no instrumental case in Latvian. Also counted as errors are instances where the wordclass does not match. For instance, the verb *atvykti* ‘arrive:INF’ is identified with the Latvian preposition *pie* ‘at’, or Lithuanian *šiol*, an adverbial form derived from a demonstrative stem occurring with the preposition *iki* in *iki šiol* ‘until now’, is identified with Latvian *līdz* ‘until’. There are two (related) single cases where there is actually a cognate stem with different wordclasses: Lithuanian *paskui* ‘after’ (preposition) and *paskos*, an adverbial genitive form occurring in *iš paskos* ‘from behind’, both go with Latvian *sekoja* ‘follow:PST3’. The *sk*-sequence in the Lithuanian forms is cognate with the verb root *sek-*.

To the set of wrong equivalents is also added a group of seven pairs where the lexical correspondence is not complete, such as Lithuanian *ežero* ‘lake:GEN:SG’ and Latvian *jūras* ‘sea:GEN:SG’ or Lithuanian *(pa) klausė* ‘ask:PST3’ and Latvian *sacīja* ‘say:PST3’.

Among the set of wrong equivalents — if we abstract from the correct cognates *paskui sekoja* — there is only one wrong alignment in case of Lithuanian *laikosi* ‘keep:PRS3:REFL’ and Latvian ***kas*** ‘who’. Cases of wrong alignment cannot be excluded, since ACT is over-generalizing; but if it is restricted by semantics (functional equivalents only) accidentally aligned pairs of forms are very rare (one of 408 in our dataset).

Table 10: ‘Correct’ and ‘wrong’ equivalents and alignment by ACT

	Aligned by ACT	Not aligned
Correct functional equivalents	406	476
No correct equivalents	2 (1 of which *paskui*)	116

Next we consider how the remaining 882 truly functionally equivalent forms are related. In cases of doubt, etymological dictionaries have been consulted (mainly Fraenkel 1962/5 and, more cautiously, Karulis 1992). The pairs are classified according to the following categories: (i) not cognate (e.g., Lithuanian *aukso*, Latvian *zelta* ‘gold:GEN:SG’), (ii) cognates both in roots and derivation (if there are any derivational affixes), (iii) cognate roots (with different derivational elements; e.g., Lithuanian *givenimas* Latvian *dzīvība* ‘life:NOM:SG’ with different nominalizing suffixes *-im-* vs. *-īb-* plus an extension with *-n-* in Lithuanian), (iv) cognate affixes but different roots (e.g., Lithuanian *at-ėjo* Latvian *at-nāca* ‘come:PST3’), and (v) parallel loans and proper names (e.g., Lithuanian *angelai* Latvian *eņģeļi* ‘angel:NOM:PL). There is a further small group of three pairs where it is unclear whether the roots are cognate (*amžinąjį mūžīgo* ‘eternal:ACC:SG:DEF’; *amžių* ‘age:GEN:PL’, *mūžos* ‘age:LOC:PL’; *amžius* ‘age:NOM:SG’ *mūžīgi* ‘eternal:ADV’). In case of a perfect result of the algorithm it is expected that all pairs of group (ii) should be aligned, and all pairs of group (i) should not be aligned. For all other groups it is expected that they should exhibit mixed behavior. Parallel loans are expected to exhibit a high number of matches as well as partial cognates where the cognate elements (roots or affixes) contain more consonants than the non-cognate elements. As shown in Table 11, this expectation is largely met. Only 36 of 306 full cognates are not identified (11.8%) and only 9 of 322 non-cognate pairs are wrongly aligned (2.8%). We will now consider these two groups of forms in detail.

Aligned but no cognates are *jam viņam* ‘he:DAT’ (with *ņ* > *j* by EC; the dative inflection *-am* is cognate); *karalystė valstība* ‘kingdom/empire:NOM’, *karalystę valstību* ‘kingdom:ACC’ (accidental coincidence of *l-s-t* sequence), *skelbti sludināt* ‘proclaim:INF’ (accidental coincidence of *s-l* sequence with shared *t* in infinitive), *stebėjosi brīnījās* 'wondered:PST3' (*j-s* matching from past and reflexive inflection, *b* accidental), *saugokitės sargieties* 'watch:IMP:2PL:REFL' (t-s matching from second plural and reflexive inflection, *s* in onset accidental), and *šviesa gaisma* 'light:NOM:SG', *šviesą gaismu* (same ACC:SG), *šviesos gaismas* (same GEN:SG) because <g> is wrongly Lithuanized to <š> (there are very few instances of <g> to <g> correspondences in the data). The few errors are thus mainly caused by cognate inflections and by a wrong result in the identification of consonants (*g*).

Table 11: Cognates and alignment by ACT

	Aligned	Not aligned	Sum
Parallel loans	75	11	86
Cognates (with derivation)	270	36	306
Cognates (roots only)	41	94	135
Possibly cognate roots	0	3	3
Cognate affixes	11	19	30
Not cognate	9	313	322
Sum	406	476	882

To reach more accuracy (= precision) it is possible to sharpen the requirements for the number of identical consonants in the ACT algorithm. In the version applied here it is only required that the two forms to be aligned have more matching consonants than any of the two forms contain non-matching consonants. However, if we sharpen the requirement such that there must be more matching consonants than the sum of non-matching consonants in the two forms to be aligned, the number of wrongly aligned non-cognate forms drops from nine to two. This version of the ACT algorithm is called ACT2. In the whole corpus the number of aligned forms drops from 408 to 363 with ACT2; in the group of cognates (with derivation) sixteen of 270 pairs will be lost. As is common in computational approaches, there is a trade-off between accuracy (= precision) and coverage (= recall; see, e.g., Cysouw *et al.* 2007), ACT2 is more accurate, but has a lower coverage.

Within the thirty-six non-aligned cognates in Table 11 we can identify two major groups. The first one concerns non-successful equivalence in consonants in the EC component and the second one is due to differences in inflectional forms. In the group cognates (with derivation) it is not assured that the inflection component is exactly cognate, it is only assured that root and derivational component (if any) are cognate.

In the EC component the first problem is solely orthographic. Latvian <dž> corresponding to Lithuanian <g> is a digraph and EC cannot identify it. Hence *gerti dzert* 'drink:INF', *girdi dzird* 'hear:PRS3', *gyvas dzīvs* 'alive.NOM:SG:M' and four other pairs are not recognized. Problems are also caused by the varying correspondences of /š/ in the two languages:

(9) Consonant correspondences involving *š* (due to sound laws or analogy)

Lith *š* Ltv *s* (8x)	Lith /s^j/ Ltv *s* (3x)	Lith *č* Ltv *š* < */t^j/ (2x)	Lith *t* Ltv *š* < */t^j/ (3x)
šaukė sauca	*duosiu došu*	*trečią trešajā*	*patį pašu*
'call:PST3'	'give:FUT1SG'	'third.ACC/LOC:DEF'	'self:ACC'

Further, there are some complications involving Lithuanian <v> and Latvian <u> (due to different reasons diachronically): *du divi* 'two', *sau sev* 'REFLEXIVE:DAT', *tau tev* 'thou:dat', *vandeniu ūdeni* 'water:INST/ACC'.

The second group of mismatches is due to differences in inflectional morphology in irregular verbs and in demonstrative pronouns: *buvo bija* 'be:PST3', *eik(ite) ej(iet)* 'go:IMP2SG(PL)' (imperative *-k* is restricted to Lithuanian), *einu eju* 'go:PRS1SG', *esu esmu* 'be:PRS1SG' (both due to loss of athematic inflection), *tą tanī* 'that:ACC:SG/LOC:SG', *tuo tanī* 'that:INST:SG/LOC:SG' (different case forms). Two further examples have only one root consonant and differ in inflection: *akių acīm* 'eye:PL. GEN/DAT', *šio šīs* 'this:GEN:SG:M/GEN.SG:F'. A special case of mismatch is the different order of the reflexive marker in verbs with prefixes: Lithuanian *pa-**si**-rodė* Latvian *pa-rādīj**ās*** 'appear:PST3'.

We have to add that for some forms it is actually very easy to align them, 35 forms are exactly identical in Lithuanian and Latvian in orthography (33 of them cognates, two names and parallel loans). However, aside of those, identifying cognates is no trivial task and it has been shown here that the proposed algorithm is quite successful, especially given that its two first components (SEF and EC) are not particularly sophisticated. The results could be considerably improved by applying ACT to root morphemes only and with loanwords and names removed and with better established consonant correspondences.

It remains to show that the algorithm works better with consonants than with vowels or with consonants and vowels. Table 12 shows the results for ACT2 (which is more accurate, as shown above). For testing the performance of vowels we simply exchange consonants and vowels. (The computer is told that the vowels are consonants and vice versa). It cannot be said that vowels are completely useless for reconstruction, but the performance of "vowel templates" instead of consonant templates is considerably weaker both in terms of accuracy and coverage even though EC does not do any bad job for vowel equivalences.

(10) Latvian to Lithuanized-Latvian vowels
a a, ā o, e e, i i, ū ū, o o, ē ė, u ų, ī y,

(11) Lithuanian to Lettonized-Lithuanian vowels
a a, ą u, i i, ū ū, į u, y ī, ų u, u u, o ā, ė ē, ę u, e e,

Treating both consonants and vowels as consonants (with no extratemplatic characters left) yields a good accuracy for strict cognates with very little else extracted, but has a considerably lower coverage than using consonant templates only.

Further evidence for the importance of consonants comes from cognate research in studies of multilingualism. Berthele (2010) investigates the performance of speakers of Germanic and Romance languages to identify words in a Germanic (Danish) or Romance (Romansh) language not familiar to them. He finds that "[i]n the listening comprehension condition consonantal contrasts (or their absence) seem to be a more important predictor for successful inferencing than vowels. If vowels are concerned, comprehension can even be better in cases where they are different. The same inverted pattern in the vowel category can be found in the reading comprehension data."

Table 12: ACT2 with consonant templates, vowel templates, and consonant-and-vowel templates

	Aligned consonants	Aligned vowels	Aligned consonants and vowels	Total (including non-aligned forms)
Parallel loans	66	28	40	86
Cognates (with derivation)	254	124	205	306
Cognates (roots only)	30	24	17	135
Possibly cognate roots	0	0	0	3
Cognate affixes	10	4	1	30
Not cognate	2	24	2	322
Not functionally equivalent	1	3	0	118
Sum	363	207	265	1000

After having evaluated the algorithm on the Lithuanian-Latvian dataset it remains to discuss in what sense this dataset is particular. First of all, the two languages are particularly well suited for comparative purposes. The sound correspondences, especially as far as consonants are concerned, are highly transparent. However, in comparison to the good sound correspondences it is astonishing that the number of cognates is not higher. Part of the explanation is that many concepts used frequently in the NT are not part of the inherited Baltic vocabulary. However, it is astonishing how many differences there are also in the core vocabulary. In many cases, both the Lithuanian and Latvian words are old inherited forms and it just happens to be the case that the two languages have retained one or the other one of the set. Such examples (from the corpus) are, for instance, Lithuanian *sūnus* Latvian *dēls* 'son', *kraujas asinis* [PL] 'blood', *vaikai bērni* 'child[PL]', *žmona sieva* 'wife', *aukso zelta* 'gold[GEN:SG]', *matyti redzēt* 'see', *arti tuvu* 'near', *ieško meklē* 'search[PRS3]', *siela dvēsele* 'soul', *greitai drīz* 'soon', *jėgos spēka* 'power[GEN.SG], *mirtis nāve* 'death', *vadinamas saukts* 'called', *su ar* 'with' and others. Not accidentally, the discussion of non-cognate etyma in the Baltic languages is a traditional topic in Baltic linguistics (see, for instance, Fraenkel 1950).

Latvian and Lithuanian are thus a pair of languages that is particularly well suited for finding cognates. In case of cognates the sound correspondences are mostly very transparent, but there are also many functional equivalents that are not cognate. Thus, there are both many cognates and many cases of lack of cognates which are not too difficult to distinguish from each other. After all — and this will not be any surprise for Baltic philologists — there is good reason why there is a bias toward historical-comparative approaches in Baltic linguistics.

4. Conclusions

It has been argued in this paper that the comparative method is highly implicit in its first step — identifying cognates. In order to make this very foundation of the comparative method more explicit, approaches in statistical natural language processing are highly useful. For an implementation on a computer an algorithm must be completely explicit otherwise it will not run automatically. Trying to formulate explicit algorithms — even if their performance cannot aspire to reach the quality of human expert labor — can help us understand what comparative linguists exactly do and it may also be useful for teaching the comparative method to people who lack the comparative linguists' intuition. Making underlying mechanisms explicit is also of crucial importance for linguistic theory.

It has been shown here that cognates can be detected with considerable accuracy fully automatically from parallel texts in a three-step procedure:

(a) Finding semantically equivalent forms (SEF)
(b) Finding equivalent consonants (EC)
(c) Alignment of the consonant template (ACT)

To begin from the end, (c) ACT relies on the idea that consonants are the skeleton of wordforms, are more stable diachronically and more likely to exhibit systematic correspondences in sound shifts. Most importantly, however, consonants are more informative than vowels. The inventory of consonants is larger in most languages than that of vowels and consonants tend to be unique in roots, which is why roots can be aligned on the basis of unique consonants (similar place of articulation avoidance in roots). Successful alignment of the consonant template, the skeleton of roots, is the precondition for identifying roots in stems, both diachronically, in finding cognates in historical linguistics, and synchronically, in assigning two ablauting or umlauting stems to the same lexeme in language acquisition. Put differently, it is argued here that identifying cognates is the same thing as successfully aligning their consonant templates.

The importance of finding sound correspondences is usually strongly emphasized in historical linguistics. A major point here is that the identification of equivalent consonants is a precondition for identifying cognates. The more important role of consonants in establishing sound correspondences remains often implicit. Campbell (2004, 127–8), for instance, does not mention it, but all examples given involve consonants.

Finally, the first step in comparative linguistics is always semantic. It must be assured that potential cognates are functionally equivalent. This is because consonant templates are not unique enough to be unique within the full wordform lexicon of languages. They are unique only within particular semantic domains. Hence, the basis for any kind of linguistic comparison, be it historical-comparative or typological, is the identification of cross-linguistic functional equivalents.

As has been shown in Section 3, the Baltic languages are particularly well suited for historical-comparative comparison, hence the bias toward historical-comparative approaches in Baltic linguistics is understandable. This should be understood in the sense of an invitation to be innovative methodologically and theoretically also in this traditional field of research.

Bernhard Wälchli
Universität Bern
Institut für Sprachwissenschaft
Längassstrasse 49, CH-3000 Bern
bernhard.waelchli@isw.unibe.ch

Abbreviations

3 — third person, ACC — accusative, ADV — adverb, DAT — dative, DEF — definite, F — feminine, FUT — future, GEN — genitive, IMP — imperative, INF — infinitive, INST — instrumental, LOC — locative, M — masculine, NOM — nominative, PL — plural, PRS — present, PST — past, REFL — reflexive, SG — singular.

References

BERTHELE, RAPHAEL. 2010. *Simple heuristics and abduction in receptive multilingualism.* Manuscript. Submitted to Applied *Linguistics Review*.

CAMPBELL, LYLE. 2004. *Historical Linguistics. An Introduction.* Edinburgh: Edinburgh University Press.

CYSOUW, MICHAEL, CHRIS BIEMANN & MATTHIAS ONGYERTH. 2007. Using Strong's Numbers in the Bible to test an automatic alignment of parallel texts. *STUF Language Typology and Universals* 60:2, 158–171.

CYSOUW, MICHAEL & HAGEN JUNG. 2007. Cognate identification and alignment using practical orthographies. *roceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, 109–116. http://www.aclweb.org/anthology-new/W/W07/W07-1314.pdf

DAHL, ÖSTEN. 2007. From questionnaires to parallel corpora in typology. *STUF Language Typology and Universals* 60:2, 172–181.

FRAENKEL, ERNST. 1962–5. *Litauisches etymologisches Wörterbuch* 1–2. (Indogermanische Bibliothek 2, Wörterbücher). Heidelberg: Winter.

FRAENKEL, ERNST. 1950. *Die baltischen Sprachen: ihre Beziehungen zu einander und zu den indogermanischen Schwesteridiomen als Einführung in die baltische Sprachwissenschaft* (Indogermanische Bibliothek 3, Untersuchungen). Heidelberg: Winter.

GOLDSMITH, JOHN. 1976. *Autosegmental Phonology.* Ph.D., MIT. (Republished New York: Garland 1979.)

GRASSMANN, HERMANN. 1863. Über die aspiraten und ihr gleichzeitiges vorhandensein im an- und auslaute von wurzeln. Ueber das urspruengliche vorhandensein von wurzeln, deren anlaut und auslaut eine aspirate enthielt. *Zeitschrift für vergleichende Sprachforschung* 12, 81–109, 110–138.

KARULIS, KONSTANTĪNS. 1992. *Latviešu etimoloģijas vārdnīca* 1–2. Rīga: Avots.

KATAMBA, FRANCIS. 1993. *Morphology*. Houndmills: Macmillan.

MANNING, CHRISTOPHER D. & HINRICH SCHÜTZE. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Mass: MIT Press.

MAYER, THOMAS, CHRISTIAN ROHRDANTZ, FRANS PLANK, PETER BAK, MIRIAM BUTT & DANIEL A. KEIM. 2010. Consonant co-occurrence in stems across languages: Automatic analysis and visualization of a phonotactic constraint. In *NLP-Ling Workshop at ACL* 2010.

MCCARTHY, JOHN. 1979. Formal Problems in Semitic Phonology and Morphology. Ph.D., MIT. (Republished New York: Garland, 1982.)

POZDNIAKOV, KONSTANTIN & GUILLAUME SEGERER. 2007. Similar place avoidance: A statistical universal. *Linguistic Typology* 11, 307–348.

SCHEMBRI, TAMARA. 2006. *The Broken Plural in Maltese*. An analysis. BA Thesis. Institute of Linguistics. University of Malta. Msida.

SUKHOTIN, BORIS V. 1962. Ėksperimental'noe vydelenie klassov bukv s pomošč'ju ĖVM. *Problemy strukturnoj lingvistiki* 234, 189–206.

WÄLCHLI, BERNHARD. Forthcoming. *Quantifying Inner Form.* Arbeitspapiere des Instituts für Sprachwissenschaft der Universität Bern.

XANTHOS, ARIS. 2007. *Apprentissage automatique de la morphologie. Le cas des structures racine–schème* Thèse présentée à la Faculté des lettres de l'Université de Lausanne.