LexStat: automatic detection of cognates in multilingual wordlists

Conference Paper · April 2012

CITATIONS
65

READS
178

LexStat: Automatic Detection of Cognates in Multilingual Wordlists

Johann-Mattis List
Institute for Romance Languages and Literature
Heinrich Heine University Düsseldorf, Germany
listm@phil.uni-duesseldorf.de

Abstract

In this paper, a new method for automatic cognate detection in multilingual wordlists will be presented. The main idea behind the method is to combine different approaches to sequence comparison in historical linguistics and evolutionary biology into a new framework which closely models the most important aspects of the comparative method. The method is implemented as a Python program and provides a convenient tool which is publicly available, easily applicable, and open for further testing and improvement. Testing the method on a large gold standard of IPA-encoded wordlists showed that its results are highly consistent and outperform previous methods.

1 Introduction

During the last two decades there has been an increasing interest in automatic approaches to historical linguistics, which is reflected in the large amount of literature on phylogenetic reconstruction (e.g. Ringe et al., 2002; Gray and Atkinson, 2003; Brown et al., 2008), statistical aspects of genetic relationship (e.g. Baxter and Manaster Ramer, 2000; Kessler, 2001; Mortarino, 2009), and phonetic alignment (e.g. Kondrak, 2002; Prokić et al., 2009; List, forthcoming).

While the supporters of these new automatic methods would certainly agree that their greatest advantage lies in the increase of repeatability and objectivity, it is interesting to note that the most crucial part of the analysis, namely the identification of cognates in lexicostatistical datasets, is still almost exclusively carried out manually. That this may be problematic was recently shown in a comparison of two large lexicostatistical datasets produced by different scholarly teams where differences in item translation and cognate judgments led to topological differences of 30% and more (Geisler and List, forthcoming). Unfortunately, automatic approaches to cognate detection still lack the precision of trained linguists' judgments. Furthermore, most of the methods that have been proposed so far only deal with bilingual as opposed to multilingual wordlists.

The LexStat method, which will be presented in the following, is a convenient tool which not only closely renders the most important aspects of manual approaches but also yields transparent decisions that can be directly compared with the results achieved by the traditional methods.

2 Identification of Cognates

2.1 The Comparative Method

In historical linguistics, cognacy is traditionally determined within the framework of the *comparative method* (Trask, 2000, 64-67). The final goal of this method is the reconstruction of proto-languages, yet the basis of the reconstruction itself rests on the identification of cognate words or morphemes within genetically related languages. Within the comparative method, cognates in a given set of language varieties are identified by applying a recursive procedure. First an initial list of putative cognate sets is created by comparing semantically and phonetically similar words from the languages to be investigated. In most of the literature dealing with the comparative method, the question of which words are most suitable for the initial compilation of cognate lists is not explicitly addressed, yet it seems obvious that the comparanda should belong to the basic vocabulary of the languages. Based on this *cognate list*, an ini-

Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH, pages 117–125,
Avignon, France, April 23 - 24 2012.

tial list of putative sound correspondences (*correspondence list*) is created. Sound correspondences are determined by aligning the cognate words and searching for sound pairs which repeatedly occur in similar positions of the presumed cognate words. After these initial steps have been made, the cognate list and the correspondence list are modified by

1. adding and deleting cognate sets from the cognate list depending on whether or not they are consistent with the correspondence list, and

2. adding and deleting sound correspondences from the correspondence list, depending on whether or not they find support in the cognate list.

These steps are repeated until the results seem satisfying enough such that no further modifications, neither of the cognate list, nor of the correspondence list, seem to be necessary.

The specific strength of the comparative method lies in the *similarity measure* which is applied for the identification of cognates: Sequence similarity is determined on the basis of *systematic sound correspondences* (Trask, 2000, 336) as opposed to similarity based on surface resemblances of phonetic segments. Thus, comparing English *token* [təʊkən] and German *Zeichen* [tsaɪçən] 'sign', the words do not really sound similar, yet their cognacy is assumed by the comparative method, since their phonetic segments can be shown to correspond regularly within other cognates of both languages.[1] Lass (1997, 130) calls this notion of similarity *genotypic* as opposed to a *phenotypic* notion of similarity, yet the most crucial aspect of correspondence-based similarity is that it is *language-specific*: Genotypic similarity is never defined in general terms but always with respect to the language systems which are being compared. Correspondence relations can therefore only be established for individual languages, they can never be taken as general statements. This may seem to be a weakness, yet it turns out that the genotypic similarity notion is one of the most crucial strengths of the comparative method: Not only does it allow us to dive deeper in the history of languages in cases where phonetic change has corrupted the former identity of cognates to such an extent that no sufficient surface similarity is left, it also makes it easier to distinguish borrowed from commonly inherited items, since the former usually come along with a greater degree of phenotypic similarity.

[1] Compare, for example, English *weak* [wiːk] vs. German *weich* [vaɪç] 'soft' for the correspondence of [k] with [ç], and English *tongue* [tʌŋ] vs. German *Zunge* [tsʊŋə] 'tongue' for the correspondence of [t] with [ts].

2.2 Automatic Approaches

In contrast to the language-specific notion of similarity that serves as the basis for cognate detection within the framework of the comparative method, most automatic methods seek to determine cognacy on the basis of surface similarity by calculating the phonetic distance or similarity between phonetic sequences (words, morphemes).

The most popular distance measures are based on the paradigm of sequence alignment. In alignment analyses two or more sequences are arranged in a matrix in such a way that all corresponding segments appear in the same column, while empty cells of the matrix, resulting from non-corresponding segments, are filled with gap symbols (Gusfield, 1997, 216). Table 1 gives an example for the alignment of German *Tochter* [tɔxtər] 'daughter' and English *daughter* [dɔːtər]: Here, all corresponding segments are inserted in the same columns, while the velar fricative [x] of the German sequence which does not have a corresponding segment in the English word is represented by a gap symbol.

German	t	ɔ	x	t	ə	r
English	d	ɔː	-	t	ə	r

Table 1: Alignment Analysis

In order to retrieve a distance or a similarity score from such an alignment analysis, the matched *residue pairs*, i.e. the segments which appear in the same column of the alignment, are compared and given a specific score depending on their similarity. How the phonetic segments are scored depends on the respective *scoring function* which is the core of all alignment analyses. Thus, the scoring function underlying the *edit distance* only distinguishes identical from non-identical segments, while the scoring function used in the ALINE algorithm of Kondrak (2002) assigns individual similarity scores for the matching of phonetic segments based on phonetic features.

Using alignment analyses, cognacy can be determined by converting the distance or similarity scores to normalized distance scores and assuming cognacy for distances beyond a certain threshold. The normalized edit distance (NED) of two sequences A and B is usually calculated by dividing the edit distance by the length of the smallest sequence. The normalized distance score of algorithms which yield similarities (such as the ALINE algorithm) can be calculated by the formula of Downey et al. (2008):

$$1 - \frac{2S_{AB}}{S_A + S_B}, \quad (1)$$

where S_A and S_B are the similarity scores of the sequences aligned with themselves, and S_{AB} is the similarity score of the alignment of both sequences. For the alignment given in Table 1, the normalized edit distance is 0.6, and the ALINE distance is 0.25.

A certain drawback of most of the common alignment methods is that their scoring function defines segment similarity on the basis of phenotypic criteria. The similarity of phonetic segments is determined on the basis of their phonetic features and not on the basis of the probability that their segments occur in a correspondence relation in genetically related languages. An alternative way to calculate phonetic similarity which comes closer to a genotypic notion of similarity is to compare phonetic sequences with respect to their *sound classes*. The concept of sound classes goes back to Dolgopolsky (1964). The original idea was "to divide sounds into such groups, that changes within the boundary of the groups are more probable than transitions from one group into another" (Burlak and Starostin, 2005, 272)[2].

In his original study, Dolgopolsky proposed ten fundamental sound classes, based on an empirical analysis of sound-correspondence frequencies in a sample of 400 languages. Cognacy between two words is determined by comparing the first two consonants of both words. If the sound classes are identical, the words are judged to be cognate. Otherwise no cognacy is assumed. Thus, given the words German *Tochter* [tɔxtər] 'daughter' and English *daughter* [dɔːtər], the sound class representation of both sequences will be TKTR and TTR, respectively. Since the first two consonants of both words do not match regarding their sound classes, the words are judged to be non-cognate.

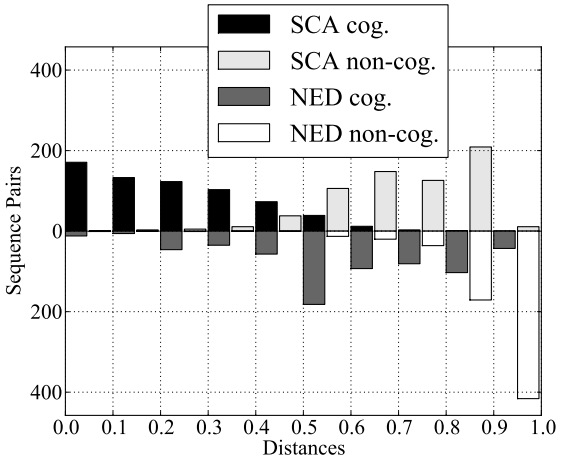

Figure 1: SCA Distance vs. NED

In recent studies, sound classes have also been used as an internal representation format for pairwise and multiple alignment analyses. The method for sound-class alignment (SCA, cf. List, forthcoming) combines the idea of sound classes with traditional alignment algorithms. In contrast to the original proposal by Dolgopolsky, SCA employs an extended sound-class model which also represents tones and vowels along with a refined scoring scheme that defines specific transition probabilities between sound classes. The benefits of the SCA distance compared to NED can be demonstrated by comparing the distance scores the methods yield for the comparison of the same data. Figure 1 contrasts the scores of NED with SCA distance for the alignment of 658 cognate and 658 non-cognate word pairs between English and German (see Sup. Mat. A). As can be seen from the figure, the scores for NED do not show a very sharp distinction between cognate and non-cognate words. Even with a "perfect" threshold of 0.8 that minimizes the number of false positive and false negative decisions there are still 13% of incorrect decisions. The SCA scores, on the other hand, show a sharper distinction between scores for cognates and non-cognates. With a threshold of 0.5 the percentage of incorrect decisions decreases to 8%.

There are only three recent approaches known to the author which explicitly deal with the task of cognate detection in multilingual wordlists. All methods take multilingual, semantically aligned

[2] My translation, original text: "[...] выделить такие группы звуков, что изменения в пределах группы более вероятны, чем переводы из одной группы в другую".

wordlists as input data. Bergsma and Kondrak (2007) first calculate the longest common subsequence ratio between all word pairs in the input data and then use an integer linear programming approach to cluster the words into cognate sets. Unfortunately, their method is only tested on a dataset containing alphabetic transcriptions; hence, no direct comparison with the method proposed in this paper is possible. Turchin et al. (2010) use the above-mentioned sound-class model and the cognate-identification criterion by Dolgopolsky (1964) to identify cognates in lexicostatistical datasets. Their method is also implemented within LexStat, and the results of a direct comparison will be reported in section 4.3. Steiner et al. (2011) propose an iterative approach which starts by clustering words into tentative cognate sets based on their alignment scores. These preliminary results are then refined by filtering words according to similar meanings, computing multiple alignments, and determining recurrent sound correspondences. The authors test their method on two large datasets. Since no gold standard for their test set is available, they only report intermediate results, and their method cannot be directly compared to the one proposed in this paper.

3 LexStat

LexStat combines the most important aspects of the comparative method with recent approaches to sequence comparison in historical linguistics and evolutionary biology. The method employs automatically extracted language-specific scoring schemes for computing distance scores from pairwise alignments of the input data. These language-specific scoring schemes come close to the notion of sound correspondences in traditional historical linguistics.

The method is implemented as a part of the LingPy library, a Python library for automatic tasks in quantitative historical linguistics.[3] It can either be used in Python scripts or directly be called from the Python prompt.

The input data are analyzed within a four-step approach: (1) sequence conversion, (2) scoring-scheme creation, (3) distance calculation, and (4) sequence clustering. In stage (1), the input sequences are converted to sound classes and their sonority profiles are determined. In stage (2), a permutation method is used to create language-specific scoring schemes for all language pairs. In stage (3) the pairwise distances between all word pairs, based on the language-specific scoring schemes, are computed. In stage (4), the sequences are clustered into cognate sets whose average distance is beyond a certain threshold.

3.1 Input and Output Format

The method takes multilingual, semantically aligned wordlists in IPA transcription as input. The input format is a CSV-representation of the way multilingual wordlists are represented in the STARLING software package for lexicostatistical analyses.[4] Thus, the input data are specified in a simple tab-delimited text file with the names of the languages in the first row, an ID for the semantic slots (*basic vocabulary items* in traditional lexicostatistic terminology) in the first column, and the language entries in the columns corresponding to the language names. The language entries should be given either in plain IPA encoding. Additionally, the file can contain headwords (items) for semantic slots corresponding to the IDs. Synonyms, i.e. multiple entries in one language for a given meaning are listed in separate rows and given the same ID. Table 2 gives an example for the possible structure of an input file.

ID	Items	German	English	Swedish
1	hand	hant	hænd	hand
2	woman	fraʊ	wʊmən	kvina
3	know	kɛnən	nəʊ	çɛna
3	know	vɪsən	–	ve:ta

Table 2: LexStat Input Format

The output format is the same as the input format except that each language column is accompanied by a column indicating the cognate judgments made by LexStat. Cognate judgments are displayed by assigning a cognate ID to each entry. If entries in the output file share the same cognate ID, they are judged to be cognate by the method.

3.2 Sequence Conversion

In the stage of sequence conversion, all input sequences are converted to sound classes, and their

[3] Online available under `http://lingulist.de/lingpy/`.

[4] Online available under `http://starling.rinet.ru/program.php`; a closer description of the software is given in Burlak and Starostin (2005, 270-275)

respective sonority profiles are calculated. LexStat uses the SCA sound-class model by default, yet other sound class models are also available.

The idea of sonority profiles was developed in List (forthcoming). It accounts for the well-known fact that certain types of sound changes are more likely to occur in specific prosodic contexts. Based on the sonority hierarchy of Geisler (1992, 30), the sound segments of phonetic sequences are assigned to different prosodic environments, depending on their prosodic context. The current version of SCA distinguishes seven different prosodic environments.[5] The information regarding sound classes and prosodic context are combined, and each input sequence is further represented as a sequence of tuples, consisting of the sound class and the prosodic environment of the respective phonetic segment. During the calculation, only those segments which are identical regarding their sound class as well as their prosodic context are treated as identical.

3.3 Scoring-Scheme Creation

In order to create language specific scoring schemes, a *permutation method* is used (Kessler, 2001). The method compares the *attested distribution* of residue pairs in phonetic alignment analyses of a given dataset to the *expected distribution*.

The attested distribution of residue pairs is derived from global and local alignment analyses of all word pairs whose distance is beyond a certain threshold. The threshold is used to reflect the fact that within the comparative method, recurrent sound correspondences are only established with respect to presumed cognate words, whereas non-cognate words or borrowings are ignored. Taking only the best-scoring word pairs for the calculation of the attested frequency distribution increases the accuracy of the approach and helps to avoid false positive matches contributing to the creation of the scoring scheme. Alignment analyses are carried out with help of the SCA method.

While the attested distribution is derived from alignments of semantically aligned words, the expected distribution is calculated by aligning word pairs without regard to semantic criteria. This is achieved by repeatedly shuffling the wordlists and aligning them with help of the same methods which were used for the calculation of the attested distributions. In the default settings, the number of repetitions is set to 1000, yet many tests showed that even the number of 100 repetitions is sufficient to yield satisfying results that do not vary significantly.

Once the attested and the expected distributions for the segments of all language pairs are calculated, a language-specific score $s_{x,y}$ for each residue pair x and y in the dataset is created using the formula

$$(2) \quad s_{x,y} = \frac{1}{r_1 + r_2} \left(r_1 \log_2 \left(\frac{a_{x,y}^2}{e_{x,y}^2} \right) + r_2 d_{x,y} \right),$$

where $a_{x,y}$ is the attested frequency of the segment pair, $e_{x,y}$ is the expected frequency, r_1 and r_2 are scaling factors, and $d_{x,y}$ is the similarity score of the original scoring function which was used to retrieve the attested and the expected distributions.

Formula (2) combines different approaches from the literature on sequence comparison in historical linguistics and biology. The idea of squaring the frequencies of attested and expected frequencies was adopted from Kessler (2001, 150), reflecting "the general intuition among linguists that the evidence of phoneme recurrence grows faster than linearly". Using the binary logarithm of the division of attested and expected frequencies of occurrence is common in evolutionary biology to retrieve similarity scores ("log-odds scores") which are apt for the computation of alignment analyses (Henikoff and Henikoff, 1992). The incorporation of the alignment scores of the original language-independent scoring-scheme copes with possible problems resulting from small wordlists: If the dataset is too small to allow the identification of recurrent sound correspondences, the language-independent alignment scores prevent the method from treating generally probable and generally improbable matchings alike. The ratio of language-specific to language-independent alignment scores is determined by the scaling factors r_1 and r_2.

As an example of the computation of language-specific scoring schemes, Table 3 shows attested and expected frequencies along with the resulting similarity scores for the matching of word-initial and word-final sound classes in the KSL testset (see Sup. Mat. B and C). The word-initial and word-final classes T = [t, d], C = [ts], S = [ʃ, s, z]

[5] The different environments are: # (word-initial, cons.), V (word-initial, vow.), C (ascending sonority, cons.), v (maximum sonority, vow.), c (descending sonority, cons.), $ (word-final, cons.), and > (word-final, vow.).

English	German	Att.	Exp.	Score
#[t,d]	#[t,d]	3.0	1.24	6.3
#[t,d]	#[ʦ]	3.0	0.38	6.0
#[t,d]	#[ʃ,s,z]	1.0	1.99	-1.5
#[θ,ð]	#[t,d]	7.0	0.72	6.3
#[θ,ð]	#[ʦ]	0.0	0.25	-1.5
#[θ,ð]	#[s,z]	0.0	1.33	0.5
[t,d]$	[t,d]$	21.0	8.86	6.3
[t,d]$	[ʦ]$	3.0	1.62	3.9
[t,d]$	[ʃ,s]$	6.0	5.30	1.5
[θ,ð]$	[t,d]$	4.0	1.14	4.8
[θ,ð]$	[ʦ]$	0.0	0.20	-1.5
[θ,ð]$	[ʃ,s]$	0.0	0.80	0.5

Table 3: Attested vs. Expected Frequencies

in German are contrasted with the word-initial and word-final sound classes T = [t, d] and D = [θ, ð] in English. As can be seen from the table, the scoring scheme correctly reflects the complex sound correspondences between English and German resulting from the High German Consonant Shift (Trask, 2000, 300-302), which is reflected in such cognate pairs as English *town* [taʊn] vs. German *Zaun* [ʦaun] 'fence', English *thorn* [θɔːn] vs. German *Dorn* [dɔrn] 'thorn', English *dale* [deɪl] vs. German *Tal* 'valley' [taːl], and English *hot* [hɔt] vs. German *heiß* [haɪs] 'hot'. The specific benefit of representing the phonetic segments as tuples consisting of their respective sound class along with their prosodic context also becomes evident: The correspondence of English [t] with German [s] is only attested in word-final position, correctly reflecting the complex change of former [t] to [s] in non-initial position in German. If it were not for the specific representation of the phonetic segments by both their sound class and their prosodic context, the evidence would be blurred.

3.4 Distance Calculation

Once the language-specific scoring scheme is computed, the distances between all word pairs are calculated. Here, LexStat uses the "end-space free variant" (Gusfield, 1997, 228) of the traditional algorithm for pairwise sequence alignments which does not penalize gaps introduced in the beginning and the end of the sequences. This modification is useful when words contain prefixes or suffixes which might distort the calculation. The alignment analysis requires no further parameters such as gap penalties, since they have already been calculated in the previous step. The similarity scores for pairwise alignments are converted to distance scores following the approach of Downey et al. (2008) which was described in section 2.2.

Word Pair			SCA	LexStat
German	*Schlange*	[ʃlaŋə]	0.44	0.67
English	*Snake*	[sneɪk]		
German	*Wald*	[valt]	0.40	0.64
English	*wood*	[wʊd]		
German	*Staub*	[ʃtaup]	0.43	0.78
English	*dust*	[dʌst]		

Table 4: SCA Distance vs. LexStat Distance

The benefits of the language-specific distance scores become obvious when comparing them with general ones. Table 4 gives some examples for non-cognate word pairs taken from the KSL testset (see Sup. Mat. B and C). While the SCA distances for these pairs are all considerably low, as it is suggested by the surface similarity of the words, the language-specific distances are all much higher, resulting from the fact that no further evidence for the matching of specific residue pairs can be found in the data.

3.5 Sequence Clustering

In the last step of the LexStat algorithm all sequences occurring in the same semantic slot are clustered into cognate sets using a flat cluster variant of the UPGMA algorithm (Sokal and Michener, 1958) which was written by the author. In contrast to traditional UPGMA clustering, this algorithm terminates when a user-defined threshold of average pairwise distances is reached.

	Ger.	Eng.	Dan.	Swe.	Dut.	Nor.
Ger. [frau]	0.00	0.95	0.81	0.70	0.34	1.00
Eng. [wʊmən]	0.95	0.00	0.78	0.90	0.80	0.80
Dan. [kvenə]	0.81	0.78	0.00	0.17	0.96	0.13
Swe. [kvinːa]	0.70	0.90	0.17	0.00	0.86	0.10
Dut. [vrɑuʊ]	0.34	0.80	0.96	0.86	0.00	0.89
Nor. [kʊinə]	1.00	0.80	0.13	0.10	0.89	0.00
Clusters	1	2	3	3	1	3

Table 5: Pairwise Distance Matrix

Table 5 shows pairwise distances of German, English, Danish, Swedish, Dutch, and Norwegian entries for the item WOMAN taken from the GER dataset (see Sup. Mat. B) along with the resulting

cluster decisions of the algorithm when setting the threshold to 0.6.

4 Evaluation

4.1 Gold Standard

In order to test the method, a gold standard was compiled by the author. The gold standard consists of 9 multilingual wordlists conforming to the input format required by LexStat (see Supplementary Material B). The data was collected from different publicly available sources. Hence, the selection of language entries as well as the manually conducted cognate judgments were carried out independently of the author. Since not all the original sources provided phonetic transcriptions of the language entries, the respective alphabetic entries were converted to IPA transcription by the author. The datasets differ regarding the treatment of borrowings. In some datasets they are explicitly marked as such and treated as non-cognates, in other datasets no explicit distinction between borrowing and cognacy is drawn. Information on the structure and the sources of the datasets is given in Table 6.

File	Family	Lng.	Itm.	Entr.	Source
GER	Germanic	7	110	814	Starostin (2008)
ROM	Romance	5	110	589	Starostin (2008)
SLV	Slavic	4	110	454	Starostin (2008)
PIE	Indo-Eur.	18	110	2057	Starostin (2008)
OUG	Uralic	21	110	2055	Starostin (2008)
BAI	Bai	9	110	1028	Wang (2006)
SIN	Sinitic	9	180	1614	Hóu (2004)
KSL	varia	8	200	1600	Kessler (2001)
JAP	Japonic	10	200	1986	Shirō (1973)

Table 6: The Gold Standard

4.2 Evaluation Measures

Bergsma and Kondrak (2007) test their method for automatic cognate detection by calculating the *set precision* (PRE), the *set recall* (REC), and the *set F-score* (FS): The set precision p is the proportion of cognate sets calculated by the method which also occurs in the gold standard. The set recall r is the proportion of cognate sets in the gold standard which are also calculated by the method, and the set F-score f is calculated by the formula

$$f = 2\frac{pr}{p+r}. \quad (3)$$

A certain drawback of these scores is that they only check for completely identical decisions regarding the clustering of words into cognate sets while neglecting similar tendencies. The similarity of decisions can be evaluated by calculating the *proportion of identical decisions* (PID) when comparing the test results with those of the gold standard. Given all pairwise decisions regarding the cognacy of word pairs inherent in the gold standard and in the testset, the differences can be displayed using a contingency table, as shown in Table 7.

	Cognate Gold Standard	**Non-Cognate Gold Standard**
Cognate Testset	true positives	false positives
Non-Cognate Testset	false negatives	true negatives

Table 7: Comparing Gold Standard and Testset

The PID score can then simply be calculated by dividing the sum of true positives and true negatives by the total number of decisions. In an analogous way the *proportion of identical positive decisions* (PIPD) and the *proportion of identical negative decisions* (PIND) can be calculated by dividing the number of true positives by the sum of true positives and false negatives, and by dividing the number of false positives by the sum of false positives and true negatives, respectively.

4.3 Results

Based on the new method for automatic cognate detection, the 9 testsets were analyzed by LexStat, using a gap penalty of -2 for the alignment analysis, a threshold of 0.7 for the creation of the attested distribution, and 1:1 as the ratio of language-specific to language-independent similarity scores. The threshold for the clustering of sequences into cognate sets was set to 0.6. In order to compare the output of LexStat with other methods, three additional analyses of the datasets were carried out: The first two analyses were based on the calculation of SCA and NED distances of all language entries. Based on these scores all words were clustered into cognate sets using the flat cluster variant of UPGMA with a threshold of 0.4 for SCA distances and a threshold of 0.7 for NED, since these both turned out to yield the best results for these approaches. The third analysis was based on the above-mentioned approach by Turchin et al. (2010). Since in this approach all decisions re-

garding cognacy are either positive or negative, no specific cluster algorithm had to be applied.

Score	LexStat	SCA	NED	Turchin
PID	0.85	0.82	0.76	0.74
PIPD	0.78	0.75	0.66	0.56
PIND	0.93	0.90	0.86	0.94
PRE	0.59	0.51	0.39	0.39
REC	0.68	0.57	0.47	0.55
FS	0.63	0.55	0.42	0.46

Table 8: Performance of the Methods

The results of the tests are summarized in Table 8. As can be seen from the table, LexStat outperforms the other methods in almost all respects, the only exception being the proportion of identical negative decisions (PIND). Since non-identical negative decisions point to false positives, this shows that – for the given settings of LexStat – the method of Turchin et al. (2010) performs best at avoiding false positive cognate judgments, but it fails to detect many cognates correctly identified by LexStat.[6] Figure 2 gives the separate PID scores for all datasets, showing that LexStat's good performance is prevalent throughout all datasets. The fact that all methods perform badly on the PIE dataset may point to problems resulting from the size of the wordlists: if the dataset is too small and the genetic distance of the languages too large, one may simply lack the evidence to prove cognacy without doubt.

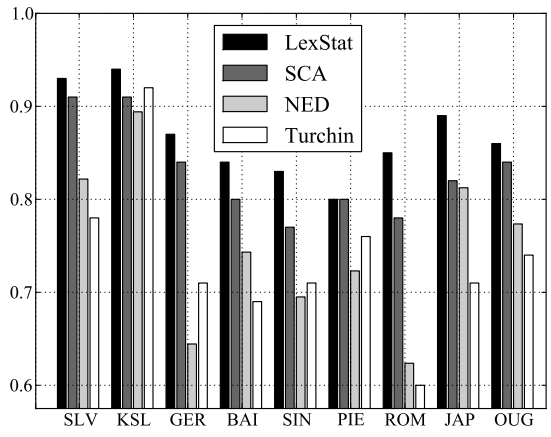

Figure 2: PID Scores of the Methods

[6]LexStat can easily be adjusted to avoid false positives by lowering the threshold for sequence clustering. Using a threshold of 0.5 will yield a PIND score of 0.96, yet the PID score will lower down to 0.82.

The LexStat method was designed to distinguish systematic from non-systematic similarities. The method should therefore produce less false positive cognate judgments resulting from chance resemblances and borrowings than the other methods. In the KSL dataset borrowings are marked along with their sources. Out of a total of 5600 word pairs, 72 exhibit a loan relation, and 83 are phonetically similar (with an NED score less then 0.6) but unrelated. Table 9 lists the number and the percentage of false positives resulting from undetected borrowings or chance resemblances for the different methods (see also Sup. Mat. D). While LexStat outperforms the other methods regarding the detection of chance resemblances, it is not particularly good at handling borrowings. LexStat cannot *per se* deal with borrowings, but only with language-specific as opposed to language-independent similarities. In order to handle borrowings, other methods (such as, e.g., the one by Nelson-Sathi et al., 2011) have to be applied.

	LexStat	SCA	NED	Turchin
Borr.	36 / 50%	44 / 61%	35 / 49%	38 / 53%
Chance R.	14 / 17%	35 / 42%	74 / 89%	26 / 31%

Table 9: Borrowings and Chance Resemblances

5 Conclusion

In this paper, a new method for automatic cognate detection in multilingual wordlists has been presented. The method differs from other approaches in so far as it employs language-specific scoring schemes which are derived with the help of improved methods for automatic alignment analyses. The test of the method on a large dataset of wordlists taken from different language families shows that it is consistent regardless of the languages being analyzed and outperforms previous approaches.

In contrast to the black box character of many automatic analyses which only yield total scores for the comparison of wordlists, the method yields transparent decisions which can be directly compared with the traditional results of the comparative method. Apart from the basic ideas of the procedure, which surely are in need of enhancement through reevaluation and modification, the most striking limit of the method lies in the data: If the wordlists are too short, certain cases of cognacy are simply impossible to be detected.

References

William H. Baxter and Alexis Manaster Ramer. 2000. Beyond lumping and splitting. Probabilistic issues in historical linguistics. In Colin Renfrew, April McMahon, and Larry Trask, editors, *Time depth in historical linguistics*, pages 167–188. McDonald Institute for Archaeological Research, Cambridge.

Shane Bergsma and Grzegorz Kondrak. 2007. Multilingual cognate identification using integer linear programming. In *RANLP Workshop on Acquisition and Management of Multilingual Lexicons*, Borovets, Bulgaria.

Cecil H. Brown, Eric W. Holman, Søren Wichmann, Viveka Velupillai, and Michael Cysouw. 2008. Automated classification of the world's languages. *Sprachtypologie und Universalienforschung*, 61(4):285–308.

Svetlana A. Burlak and Sergej A. Starostin. 2005. *Sravnitel'no-istoričeskoe jazykoznanie* [Comparative-historical linguistics]. Akademia, Moscow.

Aron B. Dolgopolsky. 1964. Gipoteza drevnejšego rodstva jazykovych semej Severnoj Evrazii s verojatnostej točky zrenija [A probabilistic hypothesis concerning the oldest relationships among the language families of Northern Eurasia]. *Voprosy Jazykoznanija*, 2:53–63.

Sean S. Downey, Brian Hallmark, Murray P. Cox, Peter Norquest, and Stephen Lansing. 2008. Computational feature-sensitive reconstruction of language relationships: Developing the ALINE distance for comparative historical linguistic reconstruction. *Journal of Quantitative Linguistics*, 15(4):340–369.

Hans Geisler and Johann-Mattis List. forthcoming. Beautiful trees on unstable ground. Notes on the data problem in lexicostatistics. In Heinrich Hettrich, editor, *Die Ausbreitung des Indogermanischen. Thesen aus Sprachwissenschaft, Archäologie und Genetik.* Reichert, Wiesbaden.

Hans Geisler. 1992. *Akzent und Lautwandel in der Romania*. Narr, Tübingen.

Russell D. Gray and Quentin D. Atkinson. 2003. Language-tree divergence times support the Anatolian theory of Indo-European origin. *Nature*, 426(6965):435–439.

Dan Gusfield. 1997. *Algorithms on strings, trees and sequences*. Cambridge University Press, Cambridge.

Steven Henikoff and Jorja G. Henikoff. 1992. Amino acid substitution matrices from protein blocks. *PNAS*, 89(22):10915–10919.

Jīng Hóu, editor. 2004. *Xiàndài Hànyǔ fāngyán yīnkù [Phonological database of Chinese dialects]*. Shànghǎi Jiàoyù, Shanghai.

Brett Kessler. 2001. *The significance of word lists. Statistical tests for investigating historical connections between languages*. CSLI Publications, Stanford.

Grzegorz Kondrak. 2002. *Algorithms for language reconstruction*. Dissertation, University of Toronto, Toronto.

Roger Lass. 1997. *Historical linguistics and language change*. Cambridge University Press, Cambridge.

Johann-Mattis List. forthcoming. SCA: Phonetic alignment based on sound classes. In Marija Slavkovik and Dan Lassiter, editors, *New directions in logic, language, and computation*. Springer, Berlin and Heidelberg.

Cinzia Mortarino. 2009. An improved statistical test for historical linguistics. *Statistical Methods and Applications*, 18(2):193–204.

Shijulal Nelson-Sathi, Johann-Mattis List, Hans Geisler, Heiner Fangerau, Russell D. Gray, William Martin, and Tal Dagan. 2011. Networks uncover hidden lexical borrowing in Indo-European language evolution. *Proceedings of the Royal Society B*, 278(1713):1794–1803.

Jelena Prokić, Martijn Wieling, and John Nerbonne. 2009. Multiple sequence alignments in linguistics. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*, pages 18–25, Stroudsburg, PA. Association for Computational Linguistics.

Donald Ringe, Tandy Warnow, and Ann Taylor. 2002. Indo-european and computational cladistics. *Transactions of the Philological Society*, 100(1):59–129.

Hattori Shirō. 1973. Japanese dialects. In Henry M. Hoenigswald and Robert H. Langacre, editors, *Diachronic, areal and typological linguistics*, pages 368–400. Mouton, The Hague and Paris.

Robert. R. Sokal and Charles. D. Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438.

George Starostin. 2008. Tower of Babel. An etymological database project. Online ressource. URL: `http://starling.rinet.ru`.

Lydia Steiner, Peter F. Stadler, and Michael Cysouw. 2011. A pipeline for computational historical linguistics. *Language Dynamics and Change*, 1(1):89–127.

Robert L. Trask, editor. 2000. *The dictionary of historical and comparative linguistics*. Edinburgh University Press, Edinburgh.

Peter Turchin, Ilja Peiros, and Murray Gell-Mann. 2010. Analyzing genetic connections between languages by matching consonant classes. *Journal of Language Relationship*, 3:117–126.

Feng Wang. 2006. *Comparison of languages in contact*. Institute of Linguistics Academia Sinica, Taipei.