Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists

Jäger, Gerhard
Tübingen University
Institute of Linguistics
Tübingen, Germany
gerhard.jaeger@uni-tuebingen.de

List, Johann-Mattis
CRLAO / AIRE
EHESS / UPMC
PARIS
info@lingpy.org

Sofroniev, Pavel
Tübingen University
Institute of Linguistics
Tübingen, Germany
pavel.sofroniev@student.uni-tuebingen.de

Abstract

Most current approaches in phylogenetic linguistics require as input multilingual word lists partitioned into sets of etymologically related words (cognates). Cognate identification is so far done manually by experts, which is time consuming and as of yet only available for a small number of well-studied language families. Automatizing this step will greatly expand the empirical scope of phylogenetic methods in linguistics, as raw wordlists (in phonetic transcription) are much easier to obtain than wordlists in which cognate words have been fully identified and annotated, even for under-studied languages. A couple of different methods have been proposed in the past, but they are either disappointing regarding their performance or not applicable to larger datasets. Here we present a new approach that uses support vector machines to unify different state-of-the-art methods for phonetic alignment and cognate detection within a single framework. Training and evaluating these method on a typologically broad collection of gold-standard data shows it to be superior to the existing state of the art.

1 Introduction

Computational historical linguistics is a relatively young sub-discipline of computational linguistics which uses computational methods to uncover how the world's 7 000 human languages have developed into their current shape. The discipline has made great strides in recent years. Exciting progress has been made with regard to automated language classification (Bowern and Atkinson, 2012; Jäger, 2015), inference regarding the time depth and geographic location of ancestral language stages (Bouckaert et al., 2012), or the identification of sound shifts and the reconstruction of ancestral word forms (Bouchard-Côté et al., 2013), to mention just a few. Most of the mentioned and related work relies on multilingual word lists manually annotated for *cognacy*. Unlike the classical NLP conception, cognate words are here understood as words in different languages which are *etymologically related*, that means, they have regularly developed from a common ancestral form, such as both English *tooth* and German *Zahn* 'tooth' that go back to an earlier Proto-Germanic word *tanθ-* with the same meaning. Manual cognate classification is a slow and labor intensive task requiring expertise in historical linguistics and intimate knowledge of the language family under investigation. From a methodological perspective, it can further be problematic to build phylogenetic inference on expert judgments, as the expert annotators necessarily base their judgments on certain hypotheses regarding the internal structure of the language family in question. In this way, the human-annotated cognate sets bear the danger of circularity. Deploying automatically inferred cognate classes thus has two advantages: it avoids the bias inherent in manually collected expert judgments and it is applicable to both well-studied and under-studied language families.

In the typical scenario, the researcher has obtained a collection of multilingual word lists in phonetic transcription (e.g. from field research or from dictionaries) and wants to classify them according to cognacy. Such datasets usually cover many languages and/or dialects (from scores to hundreds or even thousands) but only a small number of concepts (often the 200-item or 100-item Swadesh list or subsets thereof). The machine

learning task is to perform cross-linguistic clustering. There exists a growing body of gold standard data, i.e. multilingual word lists covering between 40 and 210 concepts which are manually annotated for cognacy (see Methods section for details). This suggests a supervised learning approach. The challenge here is quite different from most machine learning problems in NLP though since the goal is not to identify and deploy language-specific features based on a large amount of mono- or bi-lingual resources. Rather, the gold standard data have to be used to find cross-linguistically informative features that generalize across arbitrary language families. In the remainder of this paper we will propose such an approach, drawing on and expanding related work such as List (2014b) and Jäger and Sofroniev (2016).

2 Previous Work

Cognate detection is a *partitioning task*: a *clustering task* which does not necessarily assume a hierarchy. An early approach (Dolgopolsky, 1964) is based on the idea of *sound classes*: In order to reduce the phonetic space and to guarantee comparability across languages, sounds are clustered into classes which frequently occur in correspondence relation in genetically related languages. Dolgopolsky proposed a very rough sound class system, proposing to group all consonants into ten classes ignoring vowels. When converting all transcriptions in the data to their respective sound classes, one can use different criteria to assign words resembling each other in their sound classes to the same set of cognate words. Turchin et al. (2010) further formalized this approach and employed a modified sound class schema of 9 vowel classes to test the Altaic hypothesis. Their *Consonant Class Matching* (CCM) approach was reported to produce a low rate of false positives. Unfortunately, the rate of false negatives is also very high (List, 2014b). This is especially due to the lack of flexibility of the procedure, which hard-codes sounds to classes, ignoring that sound change is usually based on fine-grained transitions.

An alternative family of approaches to cognate detection circumvents this problem by first calculating distances or similarities between pairs of words in the data, and then feeding those scores to a flat clustering algorithm which partitions the words into cognate sets. This workflow is very common in evolutionary biology, where it is used to detect homologous genes and proteins (Bernardes et al., 2015). Two basic families of partitioning algorithms can be distinguished: hierarchical cluster algorithms and graph-based algorithms. Hierarchical cluster algorithms are based on classical agglomerative cluster algorithms (Sokal and Michener, 1958), but terminate when a user-defined threshold of average similarities among clusters is reached. In graph-based partitioning algorithms (Andreopoulos et al., 2009), words are represented as nodes in a network and links between nodes represent similarities. When clustering, links are added and removed until the nodes are partitioned into homogeneous groups (van Dongen, 2000).

More important than the clustering algorithm one uses is the computation of pairwise similarity scores between words. Here, different measures have been tested, ranging from simple string distance metrics (Bergsma and Kondrak, 2007), via enhanced sound-class-based alignment algorithms (SCA, List 2014a), to iterative frameworks in which segmental similarities between sounds are either iteratively inferred from the data (Steiner et al., 2011), or aggregated using machine learning techniques (Hauer and Kondrak, 2011). Frameworks may differ greatly regarding their underlying workflow. While the LexStat algorithm by List (2014b) uses a permutation method to compute individual segmental similarities between individual language pairs which are then fed to an alignment algorithm, the *PMI similarity approach* by Jäger (2013) infers general segmental similarities between sounds from an exhaustive parameter training procedure.

3 Materials

Benchmark data for training and testing was assembled from different previous studies and considerably enhanced by unifying semantic and phonetic representations and correcting numerous errors in the datasets. Our collection was taken from six major sources (Greenhill et al., 2008; Dunn, 2012; Wichmann and Holman, 2013; List, 2014b; List et al., 2016b; Mennecier et al., 2016)[1] and

[1]The Indo-European data from `ielex.mpi.nl` were accessed on 4-26-2016. The Austronesian data from the *Austronesian Basic Vocabulary Database* (ABVD, `language.psy.auckland.ac.nz/austronesian/`) were accessed on 12-2-2015. Among the 395 languages covered by ABVD, we only used a randomly selected subset of 100 lan-

Dataset	Words	Conc.	Lang.	Families	Cog.	Div.
ABVD (Greenhill et al. 2008)	12414	210	100	Austronesian	3558	0.27
Afrasian (Militarev 2000)	790	40	21	Afro-Asiatic	355	0.42
Bai (Wang 2006)	1028	110	9	Sino-Tibetan	285	0.19
Chinese (Hu 2004)	2789	140	15	Sino-Tibetan	1189	0.40
Chinese (Bijng Dxu 1964)	3632	179	18	Sino-Tibetan	1225	0.30
Huon (McElhanon 1967)	1176	84	14	Trans-New Guinea	537	0.41
IELex (Dunn 2012)	11479	208	52	Indo-European	2459	0.20
Japanese (Hattori 1973)	1983	199	10	Japonic	456	0.15
Kadai (Peiros 1998)	400	40	12	Tai-Kadai	103	0.17
Kamasau (Sanders 1980)	271	36	8	Torricelli	60	0.10
Lolo-Burmese (Peiros 1998)	570	40	15	Sino-Tibetan	101	0.12
Central Asian (Manni et al. 2016)	15903	183	88	Altaic (Turkic), Indo-European	895	0.05
Mayan (Brown 2008)	2841	100	30	Mayan	844	0.27
Miao-Yao (Peiros 1998)	208	36	6	Hmong-Mien	70	0.20
Mixe-Zoque (Cysouw et al. 2006)	961	100	10	Mixe-Zoque	300	0.23
Mon-Khmer (Peiros 1998)	1424	100	16	Austroasiatic	719	0.47
ObUgrian (Zhivlov 2011)	2006	110	21	Uralic	229	0.06
Tujia (Starostin 2013)	498	107	5	Sino-Tibetan	164	0.15

Table 1: Benchmark data used for the study. Items on red background were used for testing, and the rest for training. Items on white are available in ASJP transcription; all others are available in IPA transcription. The last column lists the diversity of each dataset by dividing the number of actual cognates by the number of potentially different cognates (List 2014:188).

covers datasets ranging between 100 and 210 concepts translated into 5 to 100 languages from 13 different language families.

Modifications introduced in the process of preparing the datasets included (a) the correction of errata (e.g. orthographic forms in place of phonetic representations), (b) the replacement of non-IPA symbols with their IPA counterparts (e.g. ţ → t̪ or ′ → ʔ), (c) the removal of non-IPA symbols used to convey meta-information (e.g. %), (d) removal of extraneous phonetic representation variants, and (e) the removal of morphological markers. In addition, all concept labels in the different datasets were linked to the Concepticon (`http://concepticon.clld.org`, List et al. 2016a), a resource which links concept labels to standardized concept sets in order to ease the exchange and standardization of cross-linguistic datasets. A small sample of the entries extracted from the IELex data is shown in Table 2 for illustration.

guage since the computational effort would have been impractical otherwise. For all data sets, only entries containing both a phonetic transcription and the cognate classification were used.

language	*iso*	*gloss*	*gloss_id*	*transcr.*	*cogn._class*
ELFDALIAN	qov	woman	962	ˈkɛ̀lɪŋg	woman:Ag
DUTCH	nld	woman	962	vrɑu	woman:B
GERMAN	deu	woman	962	fraŭ	woman:B
DANISH	dan	woman	962	ˈg̊ʰvenə	woman:D
DANISH_FJOLDE		woman	962	kvinʲ	woman:D
GUTNISH_LAU		woman	962	ˈkvɪnːfolk	woman:D
LATIN	lat	woman	962	ˈmulier	woman:E
LATIN	lat	woman	962	ˈfeːmina	woman:G
ENGLISH	eng	woman	962	wʊmən	woman:H
GERMAN	deu	woman	962	vaĭp	woman:H
DANISH	dan	woman	962	ˈd̥eːmə	woman:K

Table 2: Sample entries for *woman* in IELex. The cognate class identifier in the last column consists of a the concept label and an arbitrary letter combination. If two words share the same cognate class identifier, they are marked as cognate.

4 Methods

Unlike many other supervised or semi-supervised clustering tasks, the set of cluster labels to be inferred is disjoint from the gold standard labels. Therefore we chose a two-step procedure: (1) A similarity score for each pair of synonymous words from the same dataset is inferred using supervised learning, and (2) these inferred similarities are used as input for unsupervised clustering.

As for subtask (1), the relevant gold standard information are the labels "cognate" and "not cognate" for pairs of synonymous words. The sub-goal is to predict a probability distribution over these labels for unseen pairs of synonymous words. This is achieved by training a Support Vector Machine (SVM), followed by Platt scaling (Platt, 1999). The SVM primarily operates on two string similarity measure from the literature, *PMI similarity* Jäger (2013) and *LexStat similarity* (List, 2014b), which are both known to generalize well across languages and language families. We also used some auxiliary features from (Jäger and Sofroniev, 2016), which are derived from string similarities. For the clustering subtask (2), we followed List et al. (2016b) and List et al. (2017) in using the Infomap algorithm (Rosvall and Bergstrom, 2008).

The gold standard data were split into a training set and a test set. Feature selection for subtask (1) and parameter training for subtask (2) were achieved via cross-validation over the train-

ing data. For evaluation, we trained an SVM on all training data and used it to perform automatic clustering on the test data.

The remainder of this section spells out these steps in detail.

4.1 String Similarity Measures

Our strategy is to first calculate string similarities and distances between pairs of words denoting the same concept and then inferring a partition of the corresponding words from those similarities or distances via a partitioning algorithm. For word comparison we utilize two recently proposed string similarity measures.

The first string similarity measure is the one underlying the above-mentioned LexStat algorithm for automatic cognate detection (List, 2014b). The core features of the string similarity produced by the LexStat algorithm include (a) an enhanced sound-class model of 28 symbols, including tone symbols for the handling of South-East Asian tone languages, (b) a linguistically informed scoring function derived from frequently recurring directional sound change processes, and (c) a prosodic tier which automatically defines a prosodic context for each sound in a word and thus allows for a rough handling of context. The LexStat algorithm for determining string similarities can be roughly divided into four stages. In a first stage, words for the same concept in each language pair are aligned, using the SCA algorithm for phonetic alignment (List, 2014b), both globally and locally, and correspondences in the word pairs with a promising score are retained. At the same time, a randomized distribution of expected sound correspondences is calculated, using a permutation method (Kessler, 2001) in which the wordlist are shuffled, so that words denoting different concepts, which are much more likely to be not cognate, are aligned instead. In a second step, both distributions are compared, and log-odds scores (Durbin et al., 2002) for each segment pair $s_{x,y}$ are calculated (List, 2014b, 181). In a third step, the new scoring function is used to re-align the words, using a semi-global alignment algorithm which ignores prefixes or suffixes occurring in one of two strings (Durbin et al., 2002), and the similarity scores produced by classical alignment algorithms are normalized to similarity scores using the formula by Downey et al. (2008)

$$D = \frac{2 \cdot S_{AB}}{S_A + S_B} \tag{1}$$

where S_{AB} is the similarity score of an alignment of two words A and B produced by the SCA method, and S_A and S_B are the similarity scores produced by the alignment of A and B with themselves.[2]

In Jäger (2013) a data-driven method for determining string similarities is proposed which we will refer to as *PMI similarity*, as it is based on the notion of *Pointwise Mutual Information* between phonetic segments. It has successfully been used for phylogenetic inference in Jäger (2015). The method operates on phonetic strings in ASJP transcription (Brown et al., 2013) without diacritics, i.e., each segment is assigned one out of only 41 sound classes.

The PMI score of two sound classes a, b is defined as

$$\mathrm{PMI}(a, b) \doteq \log \frac{s(a, b)}{q(a)q(b)}, \tag{2}$$

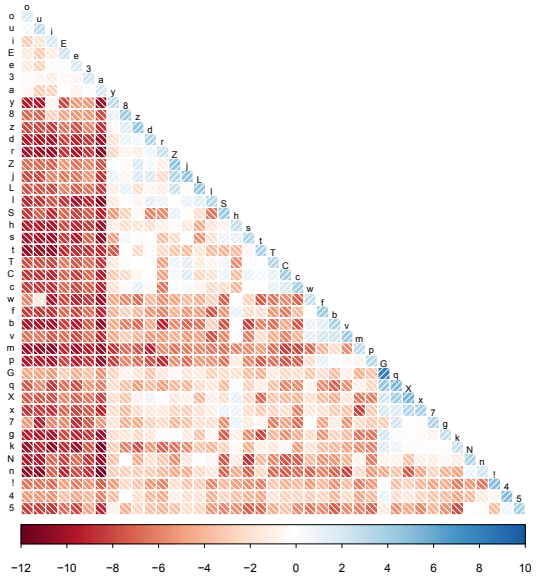
where $s(a, b)$ is the probability of a and b being aligned to each other in a pair of cognate words, and $q(a), q(b)$ are the probabilities of occurrence of a and b respectively. Sound pairs with positive PMI score provide evidence for cognacy, and vice versa.

To estimate the likelihood of sound class alignments, a corpus of *probable cognate pairs* was compiled from the ASJP data base[3] using two heuristics. First, a crude similarity measure between wordlists, based on Levenshtein distance, was defined and the 1% of all ASJP doculect[4] pairs with highest similarity were kept as *probably related*. Second, the normalized *Levenshtein distance* was computed for all translation pairs from probably related doculects. Those with a distance below a certain threshold were considered as *probably cognate*. These probable cognate pairs were used to estimate PMI scores. Subsequently, all translation pairs were aligned via the Needleman-Wunsch algorithm Needleman and Wunsch (1970) using the PMI scores from the previous step as weights. This resulted in a measure of string similarity, and all pairs above a certain similarity

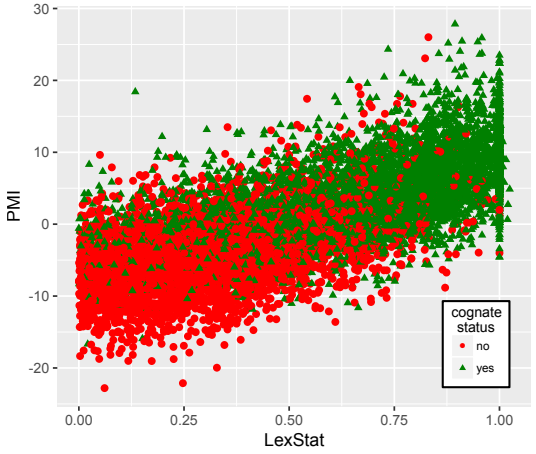
[2] The original LexStat algorithm uses distance scores by subtracting the similarity score from 1.

[3] The ASJP database Wichmann et al. (2013), available from `http://asjp.clld.org/`, is a collection of 40-item Swadesh lists from more than 6,000 languages and dialects covering all regions of the globe.

[4] *Doculect* is a neutral term for a linguistic variety which is documented in some coherent way, leaving the issue of distinguishing between languages and dialects aside.



(a) PMI scores between sound classes (Jäger 2015)



(b) Joint distribution of string similarities (IPA training data)

Figure 1: PMI scores (a) and string similarities (b) of cognate and non-cognate word pairs from the training data.

threshold were treated as probable cognates in the next step. This procedure was repeated ten times. In the last step, app. 1.3 million probable cognate pairs were used to estimate the final PMI scores.

The PMI scores thus obtained are visualized in Figure 1a (numerical values are available from the Supplementary Material of Jäger (2015)). The aggregate PMI score of a pair of aligned strings (where gaps may be inserted at any position) is defined as the sum of the PMI scores of the aligned symbol pairs. Matching a symbol with a gap incurs a penalty, with different penalties for initial and non-initial positions in a sequence of consecutive gaps.[5] The similarity $s(w_1, w_2)$ between two strings w_1, w_2 is then defined as minimal aggregate PMI score for all possible alignments. It can be computed efficiently via the Needleman-Wunsch algorithm.

There are major conceptual differences on how the two similarity measures are derived. LexStat similarity estimates separate scores between each pair of doculects, thus utilizing *regular sound correspondences*, while PMI similarity uses the same PMI scores regardless of the doculects compared. LexStat alignments further capture a prosodic tier which allows for a rough modeling of phonetic context and reflects theories on the importance of phonetic strength in sound change processes (Geisler, 1992), while the parameters used for computing PMI similarities are estimated in a purely data-driven way without using specifically linguistic insights beyond the classification of sounds into ASJP sound classes. The parameters of the PMI framework are statistically estimated using a large amount (more than 1 000 000 word pairs) of cross-linguistically diverse data. In contrast, LexStat's initial alignment algorithm is based on manually assigned parameters, and the final parameters are estimated empirically from the word pairs in the doculects being compared, and no external information is being applied. As a result, the algorithm needs a minimum of 100 concepts to yield reliable results and it yields notably better results with more than 200 words (List, 2014b; List, 2014a).

The joint distribution of LexStat and PMI string similarities for cognate and non-cognate pairs within our training set is visualized in Figure 1b.

Despite those differences, the two measures capture a similar signal; for the data from List (2014b) and List et al. (2016b), e.g., their correlation is as high as 0.727. Also, both variables are contain similar information about the binary *cognate/not cognate* variable. Figure 2 shows the *Precision-Recall curves* (cf. for instance Manning and Schütze, 1999) for LexStat and PMI similarity. While the curves are slightly different (LexStat achieves a higer precision for low recalls and PMI for high recalls), the areas under the curve are almost identical (0.893 for LexStat and 0.880 for PMI).

4.2 Workflow

In this study, we utilized both string similarity measures discussed above, as well as a collec-

[5] The values of the gap penalties were taken from Jäger (2013), where the method of estimating them is described.

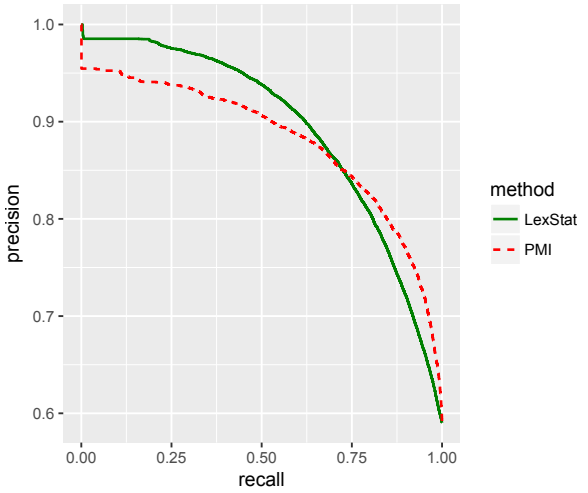


Figure 2: Precision-Recall curves for LexStat and PMI string similarities, based on the evaluation of the word pairs from the data by List (2014a).

tion of auxiliary predictors pertaining to the similarity of the doculects compared and the differential diachronic stability of lexical meanings, to infer cognate classifications. We chose a supervised learning approach using a Support Vector Machine (SVM) for this purpose. The overall workflow is shown in Figure 3. It consists of two major parts. During the first phase (the upper part in the figure shown in red), a SVM is trained on a set of training data and then used to predict the *probability of cognacy* between pairs of words from a set of test data. During the second phase (lower part in the figure, shown in green), those probabilities are used to cluster the words from the test set into *inferred cognacy classes*. The system is evaluated by comparing the inferred classification with the expert classification. We used the three largest data sets at our disposal (cf. the datasets colored in red in Table 1), *ABVD, Central Asian,* and *IELex*, for testing and all other datasets for training.

4.3 Support Vector Machine Training

Each data point during the first phase is a pair of words w_1, w_2 (i.e., a pair of phonetic strings) from doculects L_1, L_2 from data set S, both denoting the same concept c. It is mapped to a vector of values for the following features:[6]

1. LexStat string similarity between w_1 and w_2 (computed with LingPy, List and Forkel, 2016) ,
2. PMI string similarity between w_1 and w_2,
3. doculect similarity between L_1 and L_2 as defined in Jäger (2013),[7]
4. mean word length (measured in number of segments) of words for concepts c within S.
5. correlation coefficient between PMI string similarity and doculect similarity across all word pairs denoting concept c within S.[8]

The marginal distributions for cognate and non-cognate pairs of those features (for the data from List (2014b) and List et al. (2016b)) is displayed in Figure 4. It can be discerned from these plots that word length is a negative predictor and the other four features are positive predictors for cognacy.

The fact that word length is a negative predic-

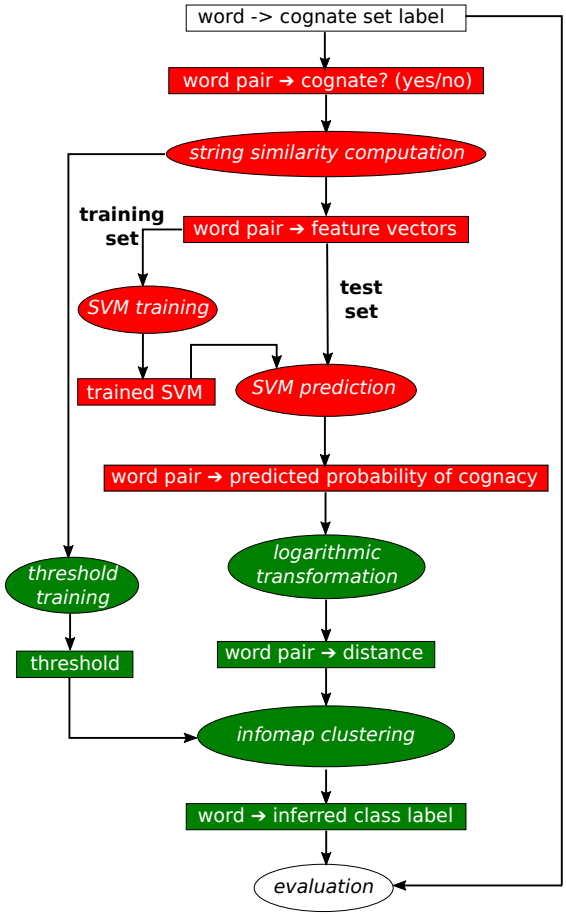


Figure 3: Workflow for supervised learning and prediction. Boxes and ellipses represent data and computations respectively.

[6]Features 2–5 are taken from (Jäger and Sofroniev, 2016). The other features used there (calibrated PMI distances and their logarithms, and the logarithm of doculect similarity) did not improve results under cross-validation over the training data.

[7]We refrain from recapitulating the full definition here for reasons of space. Essentially this amounts to the average PMI similarity between synonymous word pairs from L_1 and L_2.

[8]The last two features represent measures of the diachronic stability of concepts, based on Dellert and Buch (2016).

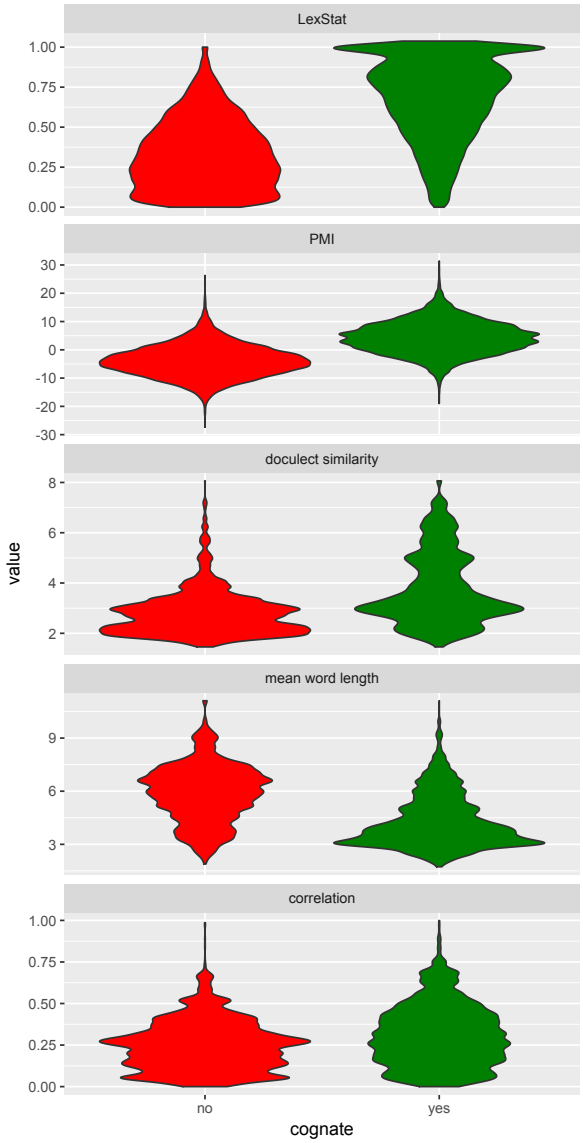


Figure 4: Distribution of features values for cognate and non-cognate word pairs

tor of cognacy arguably results from the interplay of two known regularities. (1) Pagel et al. (2007) present evidence that diachronic stability of concepts is positively correlated with their usage frequency in modern corpora. (2) According to *Zipf's Law of Abbreviation* (Zipf, 1935), there is an negative correlation between the corpus frequency of words and their lengths. Taken together, this means that concepts usually being expressed by short words tend to have a high usage frequency and therefore tend to be diachronically stable. Therefore we expect a higher proportion of cognate pairs among concepts expressed by short words than among those expressed by short words.

As the data points within the training set are mutually non-independent, we randomly chose one word pair per concept and data set for training the SVM. During the training phase, we used cross-validation over the data sets within the training set (i.e., using one training data set for validation and the other training data sets for SVM training) to identify the optimal kernel and its optimal parameters. This was carried out by completing both phases of the work flow and optimizing the *Adjusted Rand Index* (see Subsection 4.5) of the resulting classification. Training and prediction was carried out using the `svm` module from the Python package *sklearn* (`http://scikit-learn.org/stable/modules/svm.html`), which is based on the LIBSVM library (Fan et al., 2005). Predicting class membership probabilities from a trained SVM was carried out using Platt scaling (Platt, 1999) as implemented in *sklearn* (`http://scikit-learn.org`). This results in a *predicted probability of cognacy* $p(w_1, w_2|c, S)$ for each data point. The best cross-validation performance was achieved with a linear kernel with a penalty value of $C = 0.82$. Polynomial and RBF-kernels performed slightly worse. Also, we found that leaving out any subset of the features decreases performance.

4.4 Cognate Set Partitioning

In order to cluster the words into sets of potentially cognate words, we follow recent approaches by List et al. (2016b) and List et al. (2017) in using Infomap (Rosvall and Bergstrom, 2008), an algorithm which was originally designed for the detection of communities in large social networks, to detect "communities" of related words. Infomap uses random walks in undirected networks to identify the best way to assign the nodes in the network, that is, in our case, the words, to distinct groups which form a homogeneous class.

For each data set D and each concept c covered in D, a network was constructed. The vertices are all words from D denoting c. Two vertices are connected if and only if the corresponding words are predicted to be cognate with a probability $\geq \theta$ according to SVM prediction + Platt scaling. The optimal value for θ was determined as 0.66 via cross-validation over the training data. Infomap was then applied to this network, resulting in an assignment of class labels to vertices/words.

4.5 Evaluation

We used two evaluation measures to compare inferred with expert classifications on the test data.

data set	*Adjusted Rand Index*		*B-Cubed Precision*		*B-Cubed Recall*		*B-Cubed F-Score*	
	LexStat	SVM	LexStat	SVM	LexStat	SVM	LexStat	SVM
aggregated	0.676	0.683	0.868	0.847	0.838	0.869	0.850	0.855
Austronesian	0.545	0.588	0.791	0.781	0.801	0.855	0.796	0.817
Central Asian	0.866	0.843	0.916	0.883	0.962	0.981	0.938	0.929
Indo-European	0.618	0.619	0.896	0.877	0.750	0.770	0.817	0.820

Table 3: Evaluation results on the test data for the benchmark method (LexStat) and our method (SVM) according to Adjusted Rand Index and B-Cubed precision, recall, and F-score

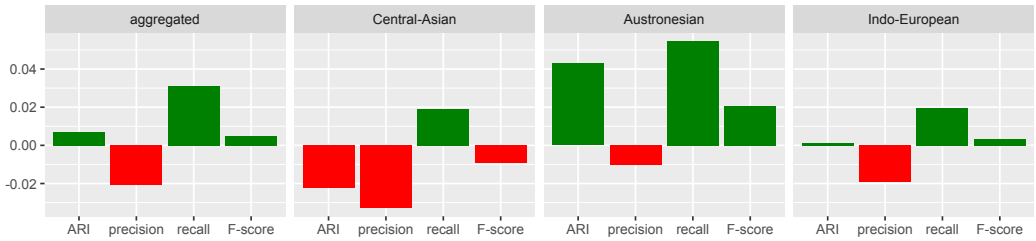


Figure 5: Evaluation results (difference between performance of our method and baseline). Green bars indicate positive values (our method outperforms baseline) and red bars indicate negative values.

The *Adjusted Rand Index* (ARI, Hubert 1985) assesses how much the equivalence relations induced by two partitions coincide. It assumes real values ≤ 1, where 1 means "perfect agreement" and 0 means "degree of agreement expected by chance". Negative values may result when from an agreement smaller than expected by chance.

B-Cubed scores (Bagga and Baldwin, 1998) measure precision and recall of a partition analysis compared against a gold standard by computing an individual accuracy score for the cluster decisions on each item in the data and then averaging the results. Hauer and Kondrak (2011) were the first to introduce this measure to test the accuracy of multilingual cognate detection algorithms. In contrast to pair scores such as ARI, B-Cubed scores have the advantage of being independent of the evaluation data itself. While pair-scores tend vary greatly depending on dataset size and cognate density, B-Cubed scores do not show this effect. They are reported as precision and recall. A low B-Cubed precision almost directly translates to the classical notion of a high amount of false positive cognate judgments made by an algorithm, while low B-Cubed recall points to a large amount of cognate sets which were missed by an algorithm.

We took the original LexStat algorithm as a baseline with which we compare our results. LexStat provides a good baseline, since it was shown to outperform alternative approaches like the above-mentioned CCM approach (Turchin et al., 2010), or clustering based on alternative string similarity measures, like the normalized edit distance, or the normalized scores of the above-mentioned SCA algorithm (List, 2014b). The LexStat implementation in LingPy offers different methods for cognate clustering. Since we employed Infomap for our SVM approach, and since Infomap clustering was shown to work well with LexStat similarities (List et al., 2017), we also used Infomap as the cluster algorithm for the LexStat approach. Since Infomap requires a threshold, we trained the threshold on our training data, excluding short wordlists. Optimal results on the training data was obtained with $\theta^* = 0.57$.

5 Results and Outlook

The evaluation results are given in Table 3, and the differences to the baseline are visualized in Figure 5. On average, the SVM-based classification shows a superior performance when compared to the baseline (an improvement of 0.7% ARI and 0.5% B-cubed F-score). This is mostly due to a substantial improvement for the Austronesian data (4.3% ARI/2.1% B-cubed F-score). Our method slightly outperforms the baseline for Indo-European but is minimally inferior when applied to the Central Asian data. While this might seem a minor improvement only, it is worth exploring on what type of data our method makes progress.

The plot in Figure 6 shows the dependency of performance (ARI) on the number concepts per data base for the training data. While this re-

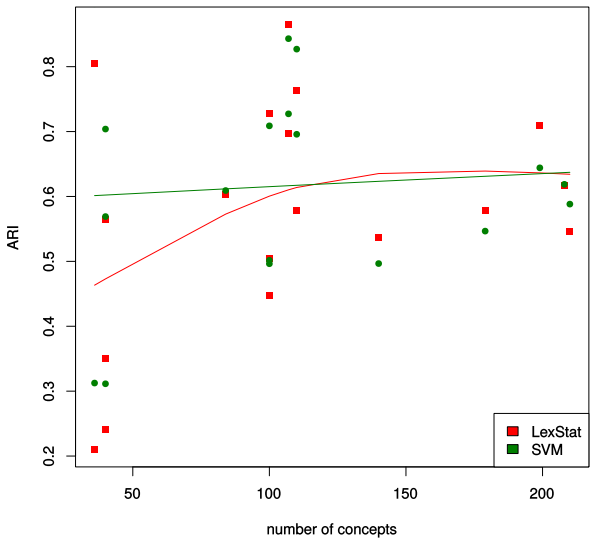


Figure 6: Performance of our method and the benchmark, depending on length of wordlists. Each dot represents one dataset/method pair. The x-axis shows the numer of concepts covered in this dataset and the y-axis the Adjusted Rand Index. Solid lines represent smoothed interpolations using Generalized Additive Models.

sult has to be taken with a grain of salt as it involves the data used for model fitting, the pattern is both plausible and striking. It shows that our method clearly outperforms LexStat if the number of concepts is smaller than 100. This finding is unsurprising since LexStat depends on regular sound correspondences. If those cannot be reliably inferred due to data sparseness, its performance drops. Our method is more robust here as it makes use of the PMI string similarity which does not rely on language-specific information. This may also explain the performance on the Austronesian data: although it covers 210 concepts across 100 languages, the languages contain many gaps, and many languages have only 100 words if not even less.

In order to get a clearer impression on where our algorithm failed, we compared false positives and negatives in the Indo-European data (Dunn, 2012), which has been investigated in deep detail during the last 200 years. While a quantitative comparison of part of speech and word length did not reveal any strong correlations with the accuracy of our approach, a qualitative analysis showed that false positives produced by our approach are usually due to *language-specific factors*. Among the factors triggering false negatives, there are specific *morphological processes* involving complex paradigms, such as Proto-Indo-European **séh$_2$wel-* 'sun', which shows many suffixes in its descendant forms, and specific instances of sound change, involving words that were drastically changed (cf. English *four* vs. French *quatre*). False positives are not only due to chance similarities (compare English *much* with Spanish *mucho*), but also due to words which share morphological elements but are marked as non-cognate in our gold standard (cf. Dutch *man* vs. German *Ehemann* 'husband'), and errors in the gold standard (cf. Upper Sorbian *powjaz* vs. Lower Sorbian *powrjoz* 'rope', wrongly marked as non-cognate in the gold standard).

The classical methods for the identification of cognate words in genetically related languages are based on the general idea that relatedness can be rigorously proven. This requires that the languages under investigation have retained enough similarity to identify regular sound correspondences. The further we go back in time, however, the less similarities we find. The fact that an algorithm like LexStat, which closely mimics the classical comparative method in historical linguistics, needs at least 100 (if not more) concepts in order to yield a satisfying performance reflects this problem of data sparseness in historical linguistics. One could argue that a serious analysis in historical linguistics should never be carried out if data are too sparse. As an alternative to this agnostic attitude, however, one could also try to work on methods that go beyond the classical framework, adding a probabilistic component, where data are too sparse to yield undisputable proof. In this paper, we have tried to make a first step into this direction by testing the power of machine learning approaches with state-of-the-art measures for string similarity in quantitative historical linguistics. The fact that our approach outperforms existing automatic approaches shows that this direction could prove fruitful in future research.

Acknowledgments

This research was supported by the ERC Advanced Grant 324246 EVOLAEMP (GJ, PS), the DFG-KFG 2237 *Words, Bones, Genes, Tools* (GJ), and the DFG research fellowship grant 261553824 *Vertical and lateral aspects of Chinese dialect history* (JML). We also thank all scholars who contributed to this study by sharing their data.

References

Bill Andreopoulos, Aijun An, Xiaogang Wang, and Michael Schroeder. 2009. A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics*, 10(3):297–314.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the ACL*, pages 79–85.

Shane Bergsma and Grzegorz Kondrak. 2007. Multilingual cognate identification using integer linear programming. In *Proceedings of the RANLP Workshop*, pages 656–663.

Juliana S. Bernardes, Fabio R. J. Vieira, Lygia M. M. Costa, and Gerson Zaverucha. 2015. Evaluation and improvements of clustering algorithms for detecting remote homologous protein families. *BMC Bioinformatics*, 16(1):1–14.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *PNAS*, 110(11):42244229.

Remco Bouckaert, Philippe Lemey, Michael Dunn, Simon J. Greenhill, Alexander V. Alekseyenko, Alexei J. Drummond, Russell D. Gray, Marc A. Suchard, and Quentin D. Atkinson. 2012. Mapping the origins and expansion of the Indo-European language family. *Science*, 337(6097):957–960, Aug.

Claire Bowern and Quentin D. Atkinson. 2012. Computational phylogenetics of the internal structure of pama-nguyan. *Language*, 88:817–845.

Cecil H Brown, Eric W Holman, Søren Wichmann, and Viveka Velupillai. 2008. Automated classification of the world s languages. *Language Typology and Universals*, 61(4):285–308.

Cecil H. Brown, Eric W. Holman, and Søren Wichmann. 2013. Sound correspondences in the world's languages. *Language*, 89(1):4–29.

Běijīng Dáxué. 1964. *Hányǔ fāngyán cíhuí* [Chinese dialect vocabularies]. Wénzì Gǎigé.

Michael Cysouw, Søren Wichmann, and David Kamholz. 2006. A critique of the separation base method for genealogical subgrouping. *Journal of Quantitative Linguistics*, 13(2-3):225–264.

Johannes Dellert and Armin Buch. 2016. Using computational criteria to extract large swadesh lists for lexicostatistics. In Christian Bentz, Gerhard Jger, and Igor Yanovich, editors, *Proceedings of the Leiden Workshop on Capturing Phylogenetic Algorithms for Linguistics*, Tübingen.

Aron B. Dolgopolsky. 1964. Gipoteza drevnejego rodstva jazykovych semej Severnoj Evrazii s verojatnostej toky zrenija. *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. *Journal of Quantitative Linguistics*, 15(4):340–369.

Michael Dunn. 2012. Indo-European lexical cognacy database (IELex). URL: `http://ielex.mpi.nl/`.

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchinson. 2002. *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, 7 edition.

Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. 2005. Working set selection using second order information for training support vector machines. *J. Mach. Learn. Res.*, 6:1889–1918, December.

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

Simon J. Greenhill, Robert Blust, and Russell D. Gray. 2008. The Austronesian Basic Vocabulary Database. *Evolutionary Bioinformatics*, 4:271–283.

Shirō Hattori. 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.

Bradley Hauer and Grzegorz Kondrak. 2011. Clustering semantically equivalent words into cognate sets in multilingual lists. In *Proceedings of the 5th International Joint NLP conference*, pages 865–873.

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

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2(1):193–218.

Gerhard Jäger and Pavel Sofroniev. 2016. Automatic cognate classification with a Support Vector Machine. In Stefanie Dipper, Friedrich Neubarth, and Heike Zinsmeister, editors, *Proceedings of the 13th Conference on Natural Language Processing*, volume 16 of *Bochumer Linguistische Arbeitsberichte*, pages 128–134. Ruhr Universität Bochum.

Gerhard Jäger. 2013. Phylogenetic inference from word lists using weighted alignment with empirical determined weights. *Language Dynamics and Change*, 3(2):245–291.

Gerhard Jäger. 2015. Support for linguistic macrofamilies from weighted alignment. *PNAS*, 112(41):1275212757.

Brett Kessler. 2001. *The significance of word lists*. CSLI Publications, Stanford.

Johann-Mattis List and Robert Forkel. 2016. *LingPy 2.5*. Max Planck Institute for the Science of Human History, Jena. URL: `http://lingpy.org`.

Johann-Mattis List, Michael Cysouw, and Robert Forkel. 2016a. *Concepticon*. Max Planck Institute for the Science of Human History, Jena. URL: `http://concepticon.clld.org`.

Johann-Mattis List, Philippe Lopez, and Eric Bapteste. 2016b. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In *Proceedings of the ACL 2016 Short Papers*, pages 599–605.

Johann-Mattis List, Simon Greenhill, and Russell Gray. 2017. The potential of automatic cognate detection for historical linguistics. *PLOS ONE*. DOI: 10.1371/journal.pone.0170046

Johann-Mattis List. 2014a. Investigating the impact of sample size on cognate detection. *Journal of Language Relationship*, 11:91–101.

Johann-Mattis List. 2014b. *Sequence comparison in historical linguistics*. Düsseldorf University Press, Düsseldorf.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, Mass.

Kenneth A. McElhanon. 1967. Preliminary observations on Huon Peninsula languages. *Oceanic Linguistics*, 6(1):1–45.

Philippe Mennecier, John Nerbonne, Evelyne Heyer, and Franz Manni. 2016. A Central Asian language survey: Collecting data, measuring relatedness and detecting loans. *Language Dynamics and Change*, 6(1).

Alexander Militarev. 2000. *Towards the chronology of Afrasian (Afroasiatic) and its daughter families*. McDonald Institute for Archaelogical Research, Cambridge.

Saul B. Needleman and Christan D. Wunsch. 1970. A gene method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, July.

Mark Pagel, Quentin D. Atkinson, and Andrew Meade. 2007. Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. *Nature*, 449(7163):717–720.

Ilia Peiros. 1998. Comparative linguistics in Southeast Asia. *Pacific Linguistics*, 142.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61–74. MIT Press.

Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *PNAS*, 105(4):1118–1123.

Joy Sanders and Arden G Sanders. 1980. Dialect survey of the Kamasau language. *Pacific Linguistics. Series A. Occasional Papers*, 56:137.

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

George S. Starostin. 2013. Annotated Swadesh wordlists for the Tujia group. In George S. Starostin, editor, *The Global Lexicostatistical Database*. RGGU, Moscow. 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.

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.

Stijn M. van Dongen. 2000. *Graph clustering by flow simulation*. PhD Thesis, University of Utrecht.

Feng Wang. 2006. *Comparison of languages in contact. The distillation method and the case of Bai*. Institute of Linguistics Academia Sinica, Taipei.

Søren Wichmann and Eric W. Holman. 2013. Languages with longer words have more lexical change. In *Approaches to measuring linguistic differences*, pages 249–281. Mouton de Gruyter, Berlin.

Søren Wichmann, André Müller, Annkathrin Wett, Viveka Velupillai, Julia Bischoffberger, Cecil H. Brown, Eric W. Holman, Sebastian Sauppe, Zarina Molochieva, Pamela Brown, Harald Hammarström, Oleg Belyaev, Johann-Mattis List, Dik Bakker, Dmitry Egorov, Matthias Urban, Robert Mailhammer, Agustina Carrizo, Matthew S. Dryer, Evgenia Korovina, David Beck, Helen Geyer, Pattie Epps, Anthony Grant, and Pilar Valenzuela. 2013. The ASJP Database. Version 16, URL: `http://asjp.clld.org`.

Mikhail Zhivlov. 2011. Annotated Swadesh wordlists for the Ob-Ugrian group. In George S. Starostin, editor, *The Global Lexicostatistical Database*. RGGU, Moscow. URL: `http://starling.rinet.ru`.

George K. Zipf. 1935. *The Psycho-Biology of Language*. MIT Press, Cambridge, Massachusetts.

Supplementary Material

The supplementary material can be downloaded from `https://zenodo.org/badge/latestdoi/77850709`. and gives all datasets used for this study along with the results, the source code needed for the replication of the study, and instructions on how to apply the software. If you find errors in the code or want to suggest improvements, please turn to our GitHub repository at `https://github.com/evolaemp/svmcc`.