# Automatic cognate classification with a Support Vector Machine

**Gerhard Jäger**
Tübingen University
Institute of Linguistics
Wilhelmstr. 19
72074 Tübingen, Germany
gerhard.jaeger@uni-tuebingen.de

**Pavel Sofroniev**
Tübingen University
Institute of Linguistics
Wilhelmstr. 19
72074 Tübingen, Germany
pavel.sofroniev@uni-tuebingen.de

## Abstract

Most current approaches in computational phylogenetic linguistics require as input multilingual word lists that are categorized into *cognate classes*. Cognate classification is currently usually done manually by experts, which is time consuming and so far 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 word lists (in phonetic transcription) are much easier to obtain than cognate-coded ones, especially for under-studied language families.

Here we propose a method for automatic cognate classification using supervised learning with a Support Vector Machine. The method outperforms Johann-Mattis List's SCA and LexStat methods (List, 2012; List, 2014b), the current *de facto* standard.

## 1 Introduction

Computational phylogenetic linguistics 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), the identification of sound shifts and the reconstruction of ancestral word forms (Bouchard-Côté et al., 2013; Hruschka et al., 2015), to mention just a few.

Most of the mentioned and related work, especially if Bayesian inference is deployed, relies on multilingual word lists that are manually annotated for cognacy (Bouchard-Côté et al., 2013, being a notable exception). Manual cognate classification is a slow and labor intensive task requiring expertise in historical linguistics and intimate knowledge of the language family under investigation. Also, building automated phylogenetic inference on expert judgments is methodologically problematic as the expert annotators necessarily base their judgments on certain hypotheses regarding the internal structure of the language family in question. In this way, certain assumptions about what is to be inferred is actually fed into the input to the inference process.

The literature contains a variety of proposals to infer cognate classifications automatically from phonetically or orthographically transcribed word lists (Kondrak, 2002; Ellison, 2007; List, 2012; Bouchard-Côté et al., 2013, *inter alia*). In the present paper we will propose a novel approach based on supervised learning. As baselines for comparison we chose List's (2012; 2014b) SCA and LexStat methods since (a) they have been tested on a variety of typologically different language families and (b) a computational implementation is freely available as part of the LingPy software package (List and Moran, 2013; List et al., 2013).

## 2 Data

We used data from five different sources:[1]

1. the benchmark data from (List, 2014a) (part of the supplementary material accompanying List 2014),
2. the annotated word lists from (Wichmann and Holman, 2013),
3. the part of the IELex data base (http://ielex.mpi.nl/, retrieved on 4-23-2013) that contains IPA transcriptions,
4. the part of the ABVD data base (Greenhill et al., 2008, see http://language.

[1] The references give the source from where we accessed the data. See the references for the ultimate sources.

psy.auckland.ac.nz/austronesian/; accessed on 12-2-2015) that contains IPA transcriptions, and

5. the Central Asian data set from (Mennecier et al., 2016).

The data from (Wichmann and Holman, 2013) are transcribed in the format of the Automated Similarity Judgment Program (ASJP; see Brown et al., 2013 for the sound class definitions). All other data are transcribed in IPA. Most datasets cover versions of a Swadesh list (see the Supplementary Material for details).

To illustrate the data format, the entries for the concept *woman* in the dataset GER from (List, 2014a) are shown in Table 1.

| *doculect* | *concept* | *transcription* | *cognate class* |
|---|---|---|---|
| Danish | woman | kvenə | 160 |
| Dutch | woman | vrɑuʋ | 158 |
| English | woman | ʋʊmən | 159 |
| German | woman | frau | 158 |
| German | woman | vaip | 159 |
| Icelandic | woman | kʰɔːna | 160 |
| Norwegian | woman | kʋinə | 160 |
| Swedish | woman | kviːa | 160 |

Table 1: Entries for *woman* in GER

Two words belong to the same cognate class if — according to historical linguistics scholarship — they descent from the same ancient proto-form.[2]

We split this collection of data bases into three parts, to be used for training (parameter estimation), validation (model selection) and testing respectively in the following way:

- **Training:** data from (List, 2014a) (except the datasets IEL and PAN, as those overlap with the validation data).

- **Validation:** data from (Wichmann and Holman, 2013).

- **Testing:** data from IELex, ABVD and (Mennecier et al., 2016).

This decision is partially motivated from practical consideration. As mentioned above, List's (2012) methods SCA and LexStat will be used as benchmark. As these methods have been developed with the data from (List, 2014a), an informative comparison should be based on the same training data. Furthermore, the data from (Wichmann and Holman, 2013) are only available in ASJP transcription. Our method uses this transcription (all IPA transcriptions are converted into ASJP format by our method), while SCA and LexStat use IPA as input. Therefore the data in ASJP format were used for model selection and the new data in IPA format were held back for final testing.

By way of a further practical consideration, LexStat, in its current implementation from LingPy, can only be applied to datasets comprising at most 169 doculects. The ABVD data comprise 395 doculects. To facilitate the comparison between methods, we split the ABVD data into four equally sized subsets.

## 3 Methods

To automatically infer cognate classes, we proceed in two steps:

- For each pair of words from the same dataset with the same meaning, the goldstandard data provide a value 0 (different cognate classes) or 1 (same cognate class). We train a binary classifier which predicts probabilities of binary class membership for each such word pair. To this end, we compute a vector of seven quantitative predictors (to be described below).

- For each group of words from the same database denoting the same concept, these pairwise probabilities are transformed into distances. The latter are used as input for hierarchical clustering, leading to an inferred cognate classification.

### 3.1 PMI similarity

In a first step, all IPA transcriptions are converted into ASJP using the converter from LingPy.

All further steps are based on the *pointwise mutual information* (PMI) between pairs of strings, using the PMI scores and gap penalties from the Supplementary Information of (Jäger, 2015); see (Jäger, 2013) for a detailed description on how those parameters are trained. PMI scores were computed as global pairwise alignment scores as implemented in the function

[2] This criterion is not always clear-cut, even if the etymology of the words involved is known. For instance, English 'woman' descends (according to the Oxford English Dictionary) from Old English 'wife+man'. Only the first of the two components is genuinely cognate with German 'Weib', so the cognacy is only partial.

`pairwise2.align.globalds` of the *Biopython* library (Cock et al., 2009).[3]

The training procedure for PMI scores between different sound classes described in Jäger (2013) ensures that pairs of different sounds frequently participating in regular sound changes have high scores. Therefore cognate word pairs tend to have high PMI similarity even if they are separated by sound changes. An example illustrating this, taken from Jäger (2015), would be the comparison of German *Hand* (`[hant]` in ASJP transcription) to its cognate, English *hand* `[hEnd]` vs. to a non-cognate such as Spanish *mano* `[mano]`. While PMI(`hant, hEnd`)$= 4.80$ since mismatches such as `a/E` and `t/d` are not very severe, PMI(`hant, mano`)$= -11.28$ since mismatches such as `h/m` and `t/o` are strongly penalized.

One reviewer suggested to use *longest common subsequence ratio* (LCSR), cf. (Melamed, 1995), or *minimum edit distance* (MED) as basic string similarity measure instead of PMI. These measure are ill-suited for cognate detection though as they both treat all non-identical sound pairs alike. To stay with the example, LCSR(`hant, hEnd`) $=$ LCSR(`hant, mano`) $= 0.5$, and MED(`hant, hEnd`) $=$ MED(`hant, mano`) $= 2$. On a more general level, the *point-biserial correlation coefficient*[4] between PMI similarity and cognacy is 0.66 for our training data, while it is only 0.58 for MED and 0.57 for LCSR. We therefore conclude that PMI similarity is a good starting point for automatic cognate identification.

Another reviewer remarked that using the same PMI parameters for all comparisons regardless of the languages involved might be sub-optimal as this does not take language-specific regular sound correspondences into account. The benchmark method LexStat does exactly that. As will be shown below, our approach still yields somewhat better results than LexStat. A thorough discussion of this important issue will have to wait for another occasion. The main reason for this discrepancy appears to be though that with the available data, language-specific parameters can be trained on 40 – 200 word pairs only, of which only a fraction is cognate and can therefore provide evidence for regular sound correspondences. This leads to a severe problem of data sparseness. The general-purpose PMI scores from (Jäger, 2015), in contradistinction, were trained on more than one million word pairs, so data sparseness is not an issue.

### 3.2 Predictors

For a given pair of words (more precisely: a pair of strings of ASJP sound classes) $w_1, w_2$ (from the same dataset), both denoting concept $c$, from doculects $D_1, D_2$, the following (dis-)similarity measures are computed:

1. **PMI similarity.**
2. **Calibrated PMI distances.** Following the procedure described in (Jäger, 2013), the PMI similarities between all pairs of non-synomymous words from $D_1, D_2$ are computed. The calibrated PMI distance between $w_1$ and $w_2$ is the relative frequency of such pairs having a higher similarity than $w_1/w_2$. This measure can be interpreted as the $p$-value for the null hypothesis that the similarity between $w_1$ and $w_2$ is due to chance. (This measure is monotonically decreasing in the previous measure; it is less fine-grained but less susceptible to chance similarities to similar sound inventories.)
3. The negative logarithm of the previous measure.
4. **Doculect similarity.** The mean value of the previous measure, averaged over all synonymous pairs from $D_1/D_2$. (This is a measure of the degree of relatedness between $D_1$ and $D_2$.)
5. The logarithm of the previous measure.
6. **Average word length.** The average length, measured in the number of ASJP symbols, of all words for concept $c$ (from the same dataset). (This is motivated by Pagel et al., 2007, — where it is shown that frequent words are more resistent against lexical replacement than rare words, together with Zipf's 1935 observation that length of words is negatively correlated with their frequency. It is therefore to be expected that stable concepts are, on average, expressed by shorter words than instable ones.)

[3] This implements a modification of the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970), disallowing a gap in one string being directly followed by a gap in the other string.

[4] The *point-biserial correlation coefficient* is a measure of the association strength between a continuous and a binary variable. It is mathematically equivalent to the Pearson correlation coefficient if the binary variable is numerically coded as 0/1.

7. **Correlation between word distance and doculect similarity.** For each pair of words for concept $c$, the correlation coefficient between their calibrated PMI distance (measure 2) and the similarity between the corresponding doculects (measure 4) is determined. (We expect this measure to be low for concepts susceptible to borrowing or sound symbolism, and to be high for stable concepts.)

Note that the first three measure quantify the (dis-)similarity between the strings $w_1/w_2$, the fourth and fifth pertain to the degree of relatedness between the doculects $D_1/D_2$, while the the last two are related to the diachronic stability of concept $c$.[5]

### 3.3 Training a binary classifier

We trained a Support Vector Machine on those vectors, using the Training Set for parameter estimation and the Validation Set for model/feature selection. As criterion to be maximized we chose the *Adjusted Rank Index* (Hubert and Arabie, 1985) as applied to the outcome of the clustering step (see below). 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 (Chang and Lin, 2011).

The test score was maximal with a Radial Basis Function kernel, a kernel coefficient $\gamma = 9 \times 10^{-4}$, and a penalty parameter $C = 0.6$ (both parameter were determined using a grid search). Leaving out any of the seven predictors led to decreased performance.

We observed that using the full collection of vectors computed from the training data led to overfitting. Generalization from the training set to the test set was improved when we randomly selected only one word pair for each data set/concept. This means that out of 111,724 word pairs from the training set, we used only 1,750 pairs (1.6%).

After training, the SVM predicts for each input vector both a categorical class label (0 or 1) and a probability distribution over class labels. Predicting class membership probabilities from a trained SVM was carried out using Platt scaling (Platt, 1999) as implemented in `http://scikit-learn.org`. In the sequel we only use the predicted probability for label 0.

### 3.4 Hierarchical clustering

For each collection of words from the same data set and denoting the same concept, the SVM predicts pairwise probabilities $p(\cdot,\cdot)$ of non-cognacy. These were transformed into pairwise distances according to the formula

$$d(w_i,w_j) \doteq \log p(w_i,w_j) - (\min_{j,k} p(w_j,w_k))$$

UPGMA clustering was performed on these distance matrices. The threshold for forming flat clusters from the UPGMA dendrogram was set at $\log 0.5 - \min_{j,k} p(w_j,w_k)$, i.e., at the distance corresponding to a 50% probability of cognacy.

## 4 Evaluation

We used two evaluation measures to determine how well an automatically inferred classification confirms to the goldstandard classification: (1) the Adjusted Rand Index (ARI), and (2) the B-Cubed score (Bagga and Baldwin, 1998).

As mentioned above, the performance of our method is compared to List's (2012; 2014b) automatic cognate classification algorithms SCA and LexStat. Perhaps the most significant difference between SCA and LexStat is that the latter automatically detects regular sound correspondences between doculects and utilizes this information to infer cognacy, while the former works with the general-purpose string similarity measures for each pair of doculects. So LexStat incorporates an important insight of the classical comparative method. Our method is closer to SCA in this respect as it also uses the same general-purpose string similarity measures for all language.

The performance of the three methods on the test set are displayed in Table 2.

We found that our method on average outperforms both LexStat and SCA. It also outperforms them for each individual data set according to both evaluation criteria, with one exception (for the Mennecier et al. data set, LexStat achieves a slightly higher B-Cubed score than our method).

## 5 Conclusion

In this short paper we demonstrated that a combination of linguistically inspired quantitative predictors, modern machine learning techniques and high-quality goldstandard training data achieves state-of-the-art performance for the recalcitrant but important task of automated cognate classification.

[5] The latter two measures are inspired by (Dellert and Buch, 2016).

| | *Adjusted Rand Index* | | | *B-Cubed score* | | |
|---|---|---|---|---|---|---|
| data set | SVM | LexStat | SCA | SVM | LexStat | SCA |
| IELex | 0.577 | 0.561 | 0.541 | 0.720 | 0.704 | 0.695 |
| Mennecier | 0.863 | 0.854 | 0.828 | 0.909 | 0.911 | 0.894 |
| ABVD-1 | 0.497 | 0.451 | 0.398 | 0.660 | 0.642 | 0.593 |
| ABVD-2 | 0.551 | 0.494 | 0.435 | 0.692 | 0.667 | 0.609 |
| ABVD-3 | 0.532 | 0.462 | 0.406 | 0.681 | 0.649 | 0.598 |
| ABVD-4 | 0.514 | 0.469 | 0.424 | 0.669 | 0.652 | 0.608 |
| weighted mean | 0.583 | 0.542 | 0.498 | 0.718 | 0.700 | 0.661 |

Table 2: Evaluation results. “SVM” refers to the method described here

These results are mostly to be understood as a proof of concept. For instance, the idea — implemented in LexStat — to utilize recurring sound correspondences for cognate identification is undoubtedly highly productive. In future research it will be explored whether more and better predictors can be inferred based on this insight.

## Acknowledgments

We thank the anonymous reviewers for KONVENS for helpful feedback and the authors of (Greenhill et al., 2008; Mennecier et al., 2016) for the kind permission to use their data. This work has been supported by the ERC Advanded Grant 324246 EVOLAEMP, which is gratefully acknowledged.

## References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 36(2):141–150.

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.

Claire Bowern and Quentin Atkinson. 2012. Computational phylogenetics and the internal structure of Pama-Nyungan. *Language*, 88(4):817–845.

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

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423. doi:10.1093/bioinformatics/btp163.

Johannes Dellert and Armin Buch. 2016. Using computational criteria to extract large Swadesh lists for lexicostatistics. ms., Tübingen.

T. Mark Ellison. 2007. Bayesian identification of cognates and correspondences. In *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, pages 15–22. Association for Computational Linguistics.

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

Daniel J. Hruschka, Simon Branford, Eric D. Smitch, Jon Wilkins, Andrew Meade, Mark Pagel, and Tanmoy Bhattachary. 2015. Detecting regular sound changes in linguistics as events of concerted evolution. *Current Biology*, 25(1):1–9.

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

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

Gerhard Jäger. 2015. Support for linguistic macrofamilies from weighted sequence alignment. *Proceedings of the National Academy of Sciences*, 112(41):12752–12757. doi: 10.1073/pnas.1500331112.

Grzegorz Kondrak. 2002. *Algorithms for Language Reconstruction*. Ph.D. thesis, University of Toronto.

Johann-Mattis List and Steven Moran. 2013. An open source toolkit for quantitative historical linguistics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Sofia, Bulgaria, August 4-9.

Johann-Mattis List, Steven Moran, Peter Bouda, and Johannes Dellert. 2013. Lingpy. Python library for automatic tasks in historical linguistics. URL: http://www.lingpy.org. Version 2.2 (Uploaded on 2013-11-22).

Johann-Mattis List. 2012. Lexstat: Automatic detection of cognates in multilingual wordlists. In Miriam Butt and Jelena Prokić, editors, *Proceedings of LINGVIS & UNCLH, Workshop at EACL 2012*, pages 117–125, Avignon.

Johann-Mattis List. 2014a. Data from: Sequence comparison in historical linguistics. GitHub Repository. Release 1.0.

Johann-Mattis List. 2014b. *Sequence Comparison in Historical Linguistics*. Düsseldorf University Press, Düsseldorf.

I. Dan Melamed. 1995. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pages 184–198, Cambridge, MA.

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). in press.

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

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.

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.

Søren Wichmann and Eric W. Holman. 2013. Languages with longer words have more lexical change. In Lars Borin and Anju Saxena, editors, *Approaches to Measuring Linguistic Differences*, pages 249–284. Mouton de Gruyter, Berlin.

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

# A Supplemental Material: Data used

| dataset | doculects | # doculects | # words | # concepts | # cognate classes | transcription |
|---|---|---|---|---|---|---|
| BAI | Bai dialects | 9 | 1,028 | 101 | 205 | IPA |
| GER | Germanic languages and dialects | 7 | 814 | 110 | 200 | IPA |
| IDS | Romance and Germanic languages | 4 | 2,429 | 550 | 1,602 | IPA |
| JAP | Japanese dialects | 10 | 1,986 | 200 | 460 | IPA |
| KSL | various languages (partially unrelated) | 7 | 1,400 | 200 | 1,208 | IPA |
| OUG | Uralic languages | 21 | 2,055 | 110 | 242 | IPA |
| PIE | Indo-European languages | 19 | 2,172 | 110 | 634 | IPA |
| ROM | Romance languages | 5 | 589 | 110 | 178 | IPA |
| SIN | Chinese dialects | 15 | 2,789 | 140 | 1,025 | IPA |
| SLV | Slavic languages | 4 | 454 | 110 | 165 | IPA |
| total | | 101 | 15,716 | 1,750 | 5,919 | |

Table 3: Data from (List, 2014a), used for training

| dataset | # doculects | # words | # concepts | # cognate classes | transcription |
|---|---|---|---|---|---|
| Afrasian | 21 | 829 | 40 | 380 | ASJP |
| Huon | 14 | 1,171 | 84 | 536 | ASJP |
| Kadai | 12 | 460 | 40 | 129 | ASJP |
| Kamasau | 8 | 271 | 36 | 60 | ASJP |
| Lolo-Burmese | 15 | 574 | 40 | 105 | ASJP |
| Mayan | 30 | 2,896 | 100 | 858 | ASJP |
| Miao-Yao | 6 | 223 | 39 | 74 | ASJP |
| Mixe-Zoque | 10 | 961 | 100 | 300 | ASJP |
| Mon-Khmer | 16 | 1,487 | 100 | 775 | ASJP |
| Moroboe | 55 | 2,040 | 138 | 582 | ASJP |
| total | 187 | 10,912 | 617 | 3,799 | |

Table 4: Data from (Wichmann and Holman, 2013), used for validation

| dataset | doculects | # doculects | # words | # concepts | # cognate classes | transcription |
|---|---|---|---|---|---|---|
| ABVD-1 | Austronesian | 99 | 14,198 | 210 | 4,592 | IPA |
| ABVD-2 | Austronesian | 99 | 14,243 | 210 | 4,156 | IPA |
| ABVD-3 | Austronesian | 99 | 13,878 | 210 | 4,181 | IPA |
| ABVD-4 | Austronesian | 98 | 14,155 | 210 | 4,435 | IPA |
| IELex | Indo-European | 55 | 8,313 | 207 | 1,998 | IPA |
| Mennecier | Central Asian | 88 | 15,904 | 183 | 895 | IPA |
| total | | 138 | 77,523 | 1,230 | 19,707 | |

Table 5: Datasets used for testing