A New Framework for Fast Automated Phonological Reconstruction Using Trimmed Alignments and Sound Correspondence Patterns

Anonymous ACL submission

Abstract

Computational approaches in historical linguistics have been increasingly applied during the past decade and many new methods that implement parts of the traditional comparative method have been proposed. Despite these increased efforts, there are not many easy-to-use and fast approaches for the task of phonological reconstruction. Here we present a new framework that combines state-of-the-art techniques for automated sequence comparison with novel techniques for phonetic alignment analysis and sound correspondence pattern detection to allow for the supervised reconstruction of word forms in ancestral languages. We test the method on a new dataset covering six groups from three different language families. The results show that our method yields promising results while at the same time being not only fast but also easy to apply and expand.

1 Introduction

Phonological reconstruction is a technique by which words in ancestral languages, which may not even be reflected in any sources, are restored through the systematic comparison of descendant words (*cognates*) in descendant languages (Fox, 1995). Traditionally, scholars apply the technique manually, but along with the recent quantitative turn in historical linguistics, scholars have increasingly tried to automate the procedure. Recent automatic approaches for linguistic reconstruction, be they supervised or unsupervised, show two major problems. First, the underlying code is rarely made publicly available, which means that they cannot be further tested by applying them to new datasets. Second, the methods have so far only been tested on a small amount of data from a limited number of language families. Thus, Bouchard-Côté et al. (2013) report remarkable results on the reconstruction of Oceanic languages, but the source code has never been published, and the method was never tested on additional datasets. Meloni et al. (2021) report very promising results for the automated reconstruction of Latin from Romance languages, using a new test set derived from a dataset originally provided by Dinu and Ciobanu (2014), but they again do not share their source code and only part of the data. Bodt and List (2021) experiment with the prediction of so far unelicited words in a small group of Sino-Tibetan languages, but they do not test the suitability of their approach for the reconstruction of ancestral languages. Jäger (2019) presents a complete pipeline by which words are clustered into cognate sets and ancestral word forms are reconstructed, but the method is only tested on a very small dataset of Romance languages.

With increasing efforts to unify and standardize lexical datasets from different sources (Forkel et al., 2018), more and more datasets that could be used to test methods for automated linguistic reconstruction have become available. Additionally, thanks to the huge progress which techniques for automated sequence comparison have made in the past decades (Kondrak, 2000; Steiner et al., 2011; List, 2014), it is much easier today to combine existing methods into new frameworks that tackle individual tasks in computational historical linguistics.

In this study, we present a new framework for automated linguistic reconstruction which combines state-of-the-art methods for automated sequence comparison with fast machine-learning techniques and test it on a newly compiled test set that covers multiple language families.

Name	Source	Subgroup	L	C	W
Bai	Wang (2004)	Bai	10	467	2892
⋆Burmish	Gong and Hill (2020)	Burmish	9	235	821
⋆Karen	Luangthongkum (2020)	Karen	11	365	3231
Lalo	Yang (2011)	Lalo (Yi)	8	1239	7522
Purus	Carvalho (2020)	Purus	4	206	724
Romance	Meloni et al. (2021)	Romance	6	4147	18806

Table 1: Datasets used in this study (L=Languages, C=Cognate Sets, W=Word Forms *=new data prepared for this study).

2 Materials

For the experiments reported here, a new cross-linguistic collection of six datasets from three language families (Sino-Tibetan, Purus, and Indo-European) was created. Data were both taken from previous sources and retro-standardized specifically for this study. The datasets, along with their sources and some basic information regarding the number of languages (L), cognate sets (C), and word forms (W) are listed in Table 1. The collection offers a rather diverse selection, in which the amount of data varies both with respect to the number of word forms, cognate sets, and languages.

3 Methods

3.1 Workflow

The new framework can be divided into a training and a prediction stage. The training consists of four steps. In step (1), the cognate sets in the training data are *aligned* with a multiple phonetic alignment algorithm. In step (2), the alignments are *trimmed* by merging sounds in the ancestral language into clusters which would leave no trace in the descendant languages (§ 3.2). In step (3), the alignments of the descendant languages are enriched by *coding for context* that might condition sound changes (§ 3.3). In step (4) the enriched alignment sites are assembled and fed to a *classifier* for training.

The prediction consists of three steps. Given a cognate set as input, the word forms are aligned with the help of the same algorithm for multiple alignment used in the training phase in step (1). In step (2), the alignment is enriched using the same method applied in the training phase and then passed to the classifier to predict the word form in the ancestral language in step (3).

Figure 1 illustrates the workflow with an example from Romance (words taken from Meloni et al. 2021). This workflow is flexible with respect to individual methods used for individual steps. For phonetic alignment, we use the Sound-Class-Based Phonetic Alignment (SCA) algorithm (List, 2012), which is the current state-of-the-art method, but any other method that yields multiple alignments could be used. The same holds for the trimming procedure, (see § 3.2), the enrichment procedure, (see § 3.3), or the classifier (see § 3.4).

3.2 Trimming Alignments

Using multiple alignments to predict ancestral or new words is nothing new and has essentially been practised by classical historical linguists for a long time (Grimm, 1822). That multiple alignments can also be used in computational frameworks has been demonstrated by List (2019a), who inferred correspondence patterns from phonetic alignments and later used these correspondence patterns to predict words missing from the data. One problem not considered in this approach, however, is that correspondence patterns can only be inferred for those cases in which descendant languages have a residue for a given sound in the ancestral language. In those cases where the sound has been lost, a prediction is not possible.

This problem is illustrated in Figure 2, where the Latin ending [ɛ] has no reflex sound in either of the descendant languages in the sample, yielding an alignment column that is completely filled with gap symbols. Our solution to deal with this problem is to post-process the multiple alignments in the training procedure by merging those columns which show only gaps in the descendant languages with the preceding alignment column. This is illustrated in Figure 3, where the Latin ending is now represented as a single sound unit [r.ɛ]. This trimming procedure is justified by the fact that correspondence patterns preceding lost sounds usually convey enough information to be distinguished from those patterns in which no sound has been lost.

3.3 Coding Context

Previous alignment-based approaches to automated word prediction have made exclusive use of the information provided by individual correspondence patterns derived from phonetic alignments (List, 2019a). While this has shown to yield already surprisingly good results, we know well that sound change often happens in certain phonetic environments. For example, we know that the initial position of a word is typically much stronger and less prone to change than the final position (Geisler, 1992). Similarly, consonants in the syllable onset position (preceding a vowel) also tend to show different types of sound change compared to consonants in the syllable offset (List, 2014). Last not least, certain sound changes may be due to "long-range dependencies", or supra-segmental features like tone, which is typically marked in the end of a morpheme in the phonetic transcription of South-East Asian languages. In order to allow a classifier to make use of this information, our framework allows to enrich the phonetic alignments further,

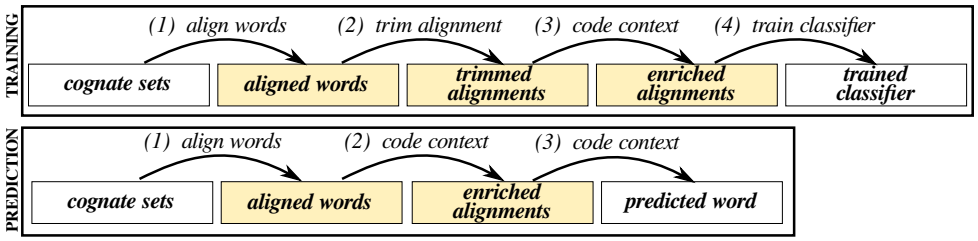

Figure 1: Workflow for the new framework for word prediction and linguistic reconstruction based on gap-free alignments and sound correspondence patterns.

	1	2	3	4	5	6	7
Latin	k	-	eː	n	aː	r	ɛ
	↑	↑	↑	↑	↑	↑	↑
Romanian	tʃ	-	i	n	a	-	-
Spanish	θ	-	e	n	a	ɾ	-
Portuguese	s	j	-	-	a	ɹ	-

Figure 2: Prediction problems when ancestral segments in multiple alignments do not show reflexes in the descendant languages.

by deriving contextual information from individual phonetic alignments and adding it to the correspondence patterns that are then used to train the classifier. An example for this procedure is given in Figure 5, where the phonetic alignment is given in traversed form, with each row corresponding to one correspondence pattern. While the information from correspondence patterns alone would only account for the first three columns of the matrix, three additional types of phonetic context have been added. Thus, column P indicates whether a pattern occurs in the beginning (^), the end ($) or the middle (−) of a word form. Column S provides information on the syllable structure following List (2014), and the remaining columns provide information on the first (*Ini*) and last (*Fin*) sound in each of the three languages, respectively. Enriching alignments should be done in a careful way, in order to avoid an over-fitting of the classifier. In our experiments, we report the results for the full coding shown in Figure 5, and contrast it with the coding including columns P and S (ignoring the initial and final sound coding), as well as the raw alignment without additional enrichment.

	1	2	3	4	5	6
Latin	k	-	eː	n	aː	r.ɛ
	↑	↑	↑	↑	↑	↑
Romanian	tʃ	-	i	n	a	-
Spanish	θ	-	e	n	a	ɾ
Portuguese	s	j	-	-	a	ɹ

Figure 3: Trimming alignments by merging sounds in the ancestral languages in those cases where an alignment column does not have sound reflexes in the descendant languages.

3.4 Classifiers

Our approach is very flexible with respect to the choice of the classifier. In order to keep the approach *fast*, we decided to restrict our experiments to the use of a Support Vector Machine (SVM) with a linear kernel, since SVMs have been successfully applied in recent approaches in computational historical linguistics dealing with different classification tasks (Jäger et al., 2017; Cristea et al., 2021). We compare this approach with the graph-based method based on correspondence patterns (henceford called CorPaR) presented by List (2019a).

3.5 Evaluation

Most scholars tend to report only the edit distance – also called Levenshtein distance (Levenshtein, 1965) – between the predicted and the attested string, both normalized by the length of the longer string and in unnormalized form. However, reporting the edit distance alone has the disadvantage that systematic differences between predicted and attested forms may be penalized too high, which is why we follow List (2019b) in computing *B-Cubed F-scores* (Amigó et al., 2009) of the alignments of source and target sequences, which measure the difference between two classifications.

3.6 Implementation

The new framework is implemented as a plugin for the LingRex Python package (List and Forkel, 2021) and allows to use classifiers from the Scikit-Learn Python package (Pedregosa et al., 2011).

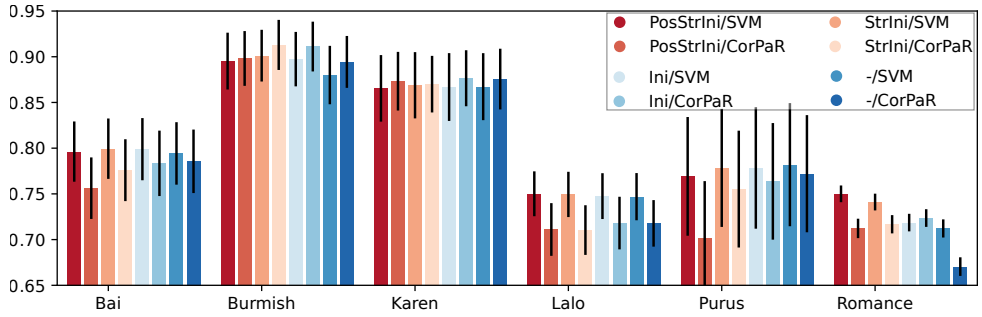

Figure 4: Comparing the results for selected coding techniques and classifiers on individual datasets.

	Ro	*Sp*	*Pt*	*P*	*S*	*Ini*		*Lt*
1	tʃ	θ	s	1	C	^	→	k
2	-	-	j	2	C	–	→	-
3	i	e	-	3	V	–	→	eː
4	n	n	-	4	C	–	→	n
5	a	a	a	5	V	–	→	aː
6	-	ɾ	ɹ	6	C	$	→	ɾ.ɛ

Figure 5: Enriching a phonetic alignment by coding various forms of context.

4 Results

In order to evaluate the framework, we tested two classifiers, a Support Vector Machine, and the CorPaR classifier (see § 3.4). Furthermore, we tested three different forms of alignment enrichment by coding individual positions of all alignment columns (`Pos`), prosodic structure (`Str`), as well as initial and final alignment columns (`IF`). For each test, we ran 100 trials in which 90% of the data were used for training and 10% for evaluation.

Table 2 shows the results for a selection of combinations between the three techniques for alignment enrichment (a full list is provided in Appendix A.2). As can be seen, the SVM classifier outperforms the CorPaR method, although the differences are not very large. While the impact of the alignment enrichment techniques on the results is not very large, we still find that they enhance the results in all SVM trials, while the raw coding of the position (`Pos`) leads to lower scores for the CorPaR classifier in our test set. For the SVM classifier, coding for prosodic structure (`Str`) and initial and final alignment columns (`StrIni`) yields the best results with respect to the normalized edit distance and the B-Cubed F-scores, while `Ini` coding outperforms the other techniques for the CorPaR classifier. From these results, we can see that alignment enrichment is a promising technique that deserves further exploration, but we do not think that the current codings are the last word on the topic.

Classifier	Analysis	ED	NED	BC
SVM	PosStrIni	0.7832	0.1656	0.8040
SVM	StrIni	0.7859	0.1648	0.8064
SVM	Str	0.7931	0.1651	0.8058
SVM	Ini	0.8171	0.1685	0.8013
SVM	none	0.8351	0.1720	0.7971
CorPaR	PosStrIni	0.8920	0.1847	0.7755
CorPaR	StrIni	0.8498	0.1758	0.7902
CorPaR	Str	0.8873	0.1773	0.7895
CorPaR	Ini	0.8242	0.1707	0.7961
CorPaR	none	0.9180	0.1819	0.7860

Table 2: Results for edit distance, normalized edit distance, and B-Cubed F-Scores on all datasets.

Figure 4 compares the results for four coding techniques on individual datasets. As can be seem from the figure, the impact of the coding techniques varies quite drastically across datasets. This shows that it would be premature to rule out any of the techniques tested here directly, but rather calls for a careful selection of alignment enrichment techniques dependent on the language family one wants to investigate.

5 Conclusion

In this study, we have presented a new framework for supervised phonological reconstruction, which is implemented in the form of a small Python package. The new framework has the advantage of being easy to use, easy to extend, and fast to apply, while at the same time yielding promising results on a newly compiled collection of datasets from three different languages families. Given that our framework can be easily extended,we hope that it will provide a solid basis for future work on phonological reconstruction in computational historical linguistics.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Timotheus Adrianus Bodt and Johann-Mattis List. 2021. Reflex prediction. a case study of western kho-bwa. *Diachronica*, 0(0):1–38.

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 of the United States of America*, 110(11):4224–4229.

Alina Maria Cristea, Liviu P. Dinu, Simona Georgescu, Mihnea-Lucian Mihai, and Ana Sabina Uban. 2021. Automatic discrimination between inherited and borrowed Latin words in Romance languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2845–2855, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fernando O. de Carvalho. 2021. A comparative reconstruction of proto-purus (arawakan) segmental phonology. *International Journal of American Linguistics*, 87(1):49–108.

Liviu Dinu and Alina Maria Ciobanu. 2014. Building a dataset of multilingual cognates for the Romanian lexicon. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1038–1043, Reykjavik, Iceland. European Language Resources Association (ELRA).

Robert Forkel, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymski, Sebastian Bank, Michael Cysouw, Harald Hammarström, Martin Haspelmath, Gereon A. Kaiping, and Russell D. Gray. 2018. Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics. *Scientific Data*, 5(180205):1–10.

Anthony Fox. 1995. *Linguistic reconstruction*. Oxford University Press, Oxford.

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

Xun Gong and Nathan Hill. 2020. *Materials for an Etymological Dictionary of Burmish*. Zenodo.

Jacob Grimm. 1822. *Deutsche Grammatik*, 2 edition, volume 1. Dieterichsche Buchhandlung, Göttingen.

Gerhard Jäger. 2019. Computational historical linguistics. *Theoretical Linguistics*, 45(3-4):151–182.

Gerhard Jäger, Johann-Mattis List, and Pavel Sofroniev. 2017. Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Long Papers*, pages 1204–1215, Valencia. Association for Computational Linguistics.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 288–295.

V. I. Levenshtein. 1965. Dvoičnye kody s ispravleniem vypadenij, vstavok i zameščenij simvolov. *Doklady Akademij Nauk SSSR*, 163(4):845–848.

Johann-Mattis List. 2012. SCA: Phonetic alignment based on sound classes. In Marija Slavkovik and Dan Lassiter, editors, *New directions in logic, language, and computation*, pages 32–51. Springer, Berlin and Heidelberg.

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

Johann-Mattis List. 2019a. Automatic inference of sound correspondence patterns across multiple languages. *Computational Linguistics*, 45(1):137–161.

Johann-Mattis List. 2019b. Beyond Edit Distances: Comparing linguistic reconstruction systems. *Theoretical Linguistics*, 45(3-4):1–10.

Johann-Mattis List and Robert Forkel. 2021. *LingRex: Linguistic reconstruction with LingPy*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Theraphan Luangthongkum. 2019. A view on proto-karen phonology and lexicon. *Journal of the Southeast Asian Linguistics Society*, 12(1):i–lii.

Carlo Meloni, Shauli Ravfogel, and Yoav Goldberg. 2021. Ab antiquo: Neural proto-language reconstruction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4460–4473, Online. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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

Feng Wang. 2004. *Language contact and language comparison. The case of Bai*. Phd, City University of Hong Kong, Hong Kong.

Cathryn Yang. 2011. *Lalo regional varieties: Phylogeny, dialectometry and sociolinguistics*. PhD dissertation, La Trobe University, Bundoora.

A Appendix

A.1 Source Code and Data

The new data collection along with the source code and the data needed to replicate the results reported in this study have been uploaded to the Open Science Framework, where they can be accessed from the link https://osf.io/myvqu/?view_only=11a008ae989f4e649743801c6734c2b1.

A.2 Table of Results (Aggregated)

Classifier	Analysis	ED	NED	BC
SVM	PosStrIni	0.7832	0.1656	0.8044
SVM	PosStr	0.7791	0.1646	0.8053
SVM	PosIni	0.8064	0.1689	0.8003
SVM	StrIni	0.7859	0.1648	0.8064
SVM	Pos	0.8002	0.1671	0.8020
SVM	Str	0.7931	0.1651	0.8058
SVM	Ini	0.8171	0.1685	0.8013
SVM	-	0.8351	0.1720	0.7971
CorPaR	PosStrIni	0.8920	0.1847	0.7755
CorPaR	PosStr	0.9050	0.1847	0.7746
CorPaR	PosIni	0.8844	0.1825	0.7772
CorPaR	StrIni	0.8498	0.1758	0.7902
CorPaR	Pos	0.9021	0.1822	0.7794
CorPaR	Str	0.8873	0.1773	0.7895
CorPaR	Ini	0.8242	0.1707	0.7961
CorPaR	-	0.9180	0.1819	0.7860

A.3 Table of Results for Individual Datasets

A.3.1 SVM

DATASET	PosStrIni	StrIni	Str	Ini	-
Bai	0.7963	0.7994	0.7976	0.7989	0.7942
Burmish	0.8952	0.9012	0.8994	0.8974	0.8800
Karen	0.8654	0.8688	0.8709	0.8669	0.8673
Lalo	0.7501	0.7494	0.7493	0.7475	0.7470
Purus	0.7691	0.7784	0.7847	0.7784	0.7819
Romance	0.7501	0.7411	0.7328	0.7186	0.7122

A.3.2 CorPaR

DATASET	PosStrIni	StrIni	Str	Ini	-
Bai	0.7562	0.7759	0.7767	0.7834	0.7856
Burmish	0.8981	0.9129	0.9127	0.9111	0.8944
Karen	0.8733	0.8700	0.8783	0.8765	0.8756
Lalo	0.7111	0.7104	0.7154	0.7181	0.7177
Purus	0.7021	0.7552	0.7680	0.7637	0.7721
Romance	0.7122	0.7168	0.6859	0.7236	0.6705