IPA alignment using vector representations

Pavel Sofroniev
author
pavel.sofroniev@student.uni-tuebingen.de

Çağri Çöltekin
advisor
ccoltekin@sfs.uni-tuebingen.de

March 2018

Abstract

This paper explores an alternative approach to pairwise alignment of IPA-encoded sound sequences in the context of computational historical linguistics: employing the cosine distance between representations of IPA segments in vector space as the scoring function in the standard sequence alignment algorithm. We implement and evaluate several different methods for obtaining such IPA embeddings, including three data-driven methods. Our experiments suggest that data-driven embeddings perform better than their counterparts grounded in general linguistics theory; and, for data-driven methods, that modelling the directionality of context yields better-performing vector representations.

Chapter 1

Introduction

Most of the computational methods developed in the field of historical linguistics involve the task of aligning sound sequences, either on its own or as a necessary step in a larger application. In its essence, sequence alignment is a way to arrange two or more sequences together in order to identify sub-sequences which are similar according to certain pre-defined criteria. Usually in the context of historical linguistics these are sequences of phonological or phonetic segments comprising transcriptions of words; and the similarity of given segments is subsequently used to infer either the cognacy of these words or rules of sound correspondences.

The currently standard algorithm for pairwise (i.e. aligning two sequences at a time) alignment was originally developed by Needleman and Wunsch (1970) for aligning amino acid sequences. Given a function that assigns scores to pairs of sequence elements, the Needleman-Wunsch algorithm is guaranteed to find the alignment(s) that minimises/maximises the overall score. There exist a number of variants of the basic algorithm, e.g. Smith and Waterman (1981), and many of these have been adapted for the purposes of computational linguistics; however, the output always depends to a great extent on the scoring function. Some of these variants have been developed for aligning more than two sequences at once, e.g. the T-Coffee algorithm by Notredame et al. (2000), but in this paper we focus on pairwise alignment only.

Unlike in bioinformatics, which deals primarily with sequences composed out of small sets of well-defined units, in historical linguistics the encoding of sequences posed for alignment is not a solved problem. Even though in general the standard way to encode sound sequences is IPA, the International Phonetic Alphabet, for the purposes of alignment the overwhelming diversity of sound segments impedes the creation of a useful scoring function. That is why modern sound sequence alignment methods tend to operate on small sets of segments; IPA-encoded data is handled by translating it into the respective reduced alphabet. While this constitutes a valid approach to tackle the problem of deriving a scoring function for the rich set of IPA segments, in this paper we experiment with an alternative approach: vectorisation. The goal is to obtain representations of IPA segments in a vector space in which the cosine similarity between segments' vectors would reflect the segments' similarity and, hence, probability of shared ancestry.

Our inspiration for experimenting with vector representations (also known

as embeddings) is drawn from the success of such methods in other fields of computational linguistics: e.g. Mikolov et al. (2013) affirm that vector representations of words exhibit useful semantic and syntactic properties; and Chen and Manning (2014) use word embeddings for transition-based parsing. More closely related to our topic, Silfverberg et al. (2018) obtain vector representations of orthographic segments for Finnish, Spanish and Turkish from recurrent neural networks trained to perform an inflection task.

1.1 Background

This section provides a concise overview of sound sequence alignment methods reported in the computational historical linguistics literature.

1.1.1 Early methods

An early method for sound sequence alignment reported in the literature is the one proposed by Covington (1996), which was subsequently improved and extended in Covington (1998). Although the method does not take advantage of any standard sequence alignment algorithm and instead performs branch-and-bound search, it does use a scoring function. This function assigns hand-crafted values depending on whether the segments are consonants, vowels, or glides. Due to this simple classification, the method could be adapted to work with IPA sequences (the author uses an undocumented encoding); however, its poor performance in comparison to more complex methods demonstrates that such coarse three-way classification is not sufficient.

ALINE, first introduced by Kondrak (2000) and further elaborated in Kondrak and Hirst (2002) and Kondrak (2003), employs a complex scoring function that is built around multi-valued (e.g. manner of articulation) and binary (e.g. nasality) phonetic features, each with its relative weight, as well as some additional parameters (e.g. relative weight of vowels). While some of these numerical values have been determined based on observations from the field of articulatory phonetics, others have been established by trial and error.

1.1.2 ASJP

The Automated Similarity Judgement Program (ASJP) is an actively developed database aiming to provide the translations of a set of 40 basic concepts into all the world's languages. At the time of writing the database covers 294 548 words from 7221 languages (Wichmann et al., 2016). The words are encoded using a uniform transcription consisting of 34 consonants and 7 vowels, introduced by Brown et al. (2008) and further specified by Brown et al. (2013) as ASJPcode, but more commonly referred to as ASJP. This restricted set of encoding symbols results in much of the phonetic information being lost, but the authors argue that such information might not be that valuable for certain applications. The simplified encoding also allows for a greater reach of the database as many of the world's languages lack sufficiently detailed phonetic record.

Jäger (2013) uses the ASJP database to calculate the partial mutual information (PMI) scores — a logarithmic measure for sound similarity based on occurrence and co-occurrence of the sound segments in data — for each pair of

ASJP segments. These PMI scores are successfully employed for inferring phylogenetic trees through determining language distances from PMI-based sequence alignment. Unlike the aforementioned methods, this one obtains its scoring function in a data-driven manner, an approach also embraced in our paper.

PMI-based scoring has been used on IPA-encoded datasets as well, by converting the sequences to ASJP as a pre-processing step. As an example, Jäger and Sofroniev (2016) use this approach to train a support vector machine (SVM) for cognate classification.

1.1.3 Other methods

The sound-class-based phonetic alignment (SCA) method developed by List (2012) employs a set of 28 sound classes. It operates on IPA sequences by converting the segments into their respective sound classes, aligning the sound class tokens, and then converting these back into IPA. The scoring function is hand-crafted to reflect the perceived probabilities of sound change transforming a segment of one class into a segment of another.

Chapter 2

Methods

We consider 5 different methods for obtaining vector representations of IPA segments. With all methods, the representations are employed for computing the cosine distance between IPA segments (cf. Figure 2.1), which comprises the scoring function in an otherwise standard application of the Needleman-Wunsch algorithm.

$$f(u,v) = 1 - \cos(\alpha) = 1 - \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

Figure 2.1: The scoring function; u and v are vector representations of two IPA segments; α is the angle between u and v

The code implementation of the complete methods, as well as of the experiments described in the next chapter, is open sourced under the GNU GPLv3 license and publicly available online.[1] It is written using the Python programming language and a set of common dependencies, including NumPy (Walt et al., 2011), SciPy (Jones et al., 2001), scikit-learn (Pedregosa et al., 2011), Gensim (Řehůřek and Sojka, 2010), and Keras (Chollet et al., 2015).

2.1 One-hot encoding

Under one-hot encoding, given a vocabulary of N distinct segments, each segment would be represented as a distinct binary vector of size N, such that exactly one of its dimensions has value 1 and all its other dimensions have value 0. In such a vector space, all vectors have length 1 and any two non-identical vectors are orthogonal.

One-hot encoding is a simple method, both conceptually and computationally, but it cannot be very useful for producing distance measures because un-

[1] `https://github.com/pavelsof/ipavec`

der its model each segment is equidistant from all the others; i.e. the method is equivalent to treating each IPA segment as a basic symbol with no connection to any other IPA segment. For the purposes of our study one-hot vector representations are used as a baseline for comparing other methods.

2.2 PHOIBLE feature vectors

PHOIBLE Online is an ongoing project aiming to compile a comprehensive tertiary database of the world languages' phonological inventories. At the time of writing the database contains phonological inventories for 1672 distinct languages making use of 2160 distinct IPA segments (Moran et al., 2014).

As part of the project the developers are also maintaining a table of phonological features, effectively mapping each segment encountered in the database to a unique ternary feature vector (values indicate either the presence, absence, or non-applicability of the respective feature). The feature table includes 39 distinct features and is based on research by Hayes (2009) and Moisik and Esling (2011). Table 2.1 lists these, grouped by value, for an arbitrary segment.

	present	absent	not applicable
ə	syllabic, sonorant, continuant, approximant, dorsal, periodic glottal source	stress, short, long, consonantal, tap, trill, nasal, lateral, labial, coronal, high, low, front, back, tense, retracted tongue root, advanced tongue root, epilaryngeal source, spread glottis, constricted glottis, raised larynx ejective, lowered larynx implosive	tone, delayed release, round, labiodental, anterior, distributed, strident, fortis, click

Table 2.1: The PHOIBLE features grouped by value for the segment ə

The PHOIBLE feature vectors comprise segment representations grounded in phonology and linguistics theory that could be readily used for a variety of computational endeavours. Nevertheless, cosine distances in the vector space defined by the PHOIBLE features is equally affected by both common and rare features/dimensions; for example the presence of a feature such as advanced tongue root in one of a pair of vectors could result in a distance larger than needed for the purposes of alignment. That is why we have also experimented with reducing the number of dimensions by applying principal component analysis (PCA), expecting that this would decrease the influence of rare features; we refer to this method as PHOIBLE-PC.

2.3 phon2vec embeddings

The first data-driven method we consider is word2vec, developed by Mikolov et al. (2013). It comprises two closely related model architectures for computing

vector representations of words: the continuous bag-of-words model (CBOW) which tries to predict a word based on its context of neighbouring words, and the continuous skip-gram model which, given a word, tries to predict words from its context. The resulting embeddings also depend on a few other hyperparameters, e.g. the number of neighbouring words that constitute a word's context, or the number of negative samples for each positive sample.

As our study concerns IPA segments rather than words, we refer to the method as phon2vec. Our embeddings are trained on a dataset of tokenised IPA sequences, with the model effectively treating the IPA segments as words and the sequences as sentences.

2.4 NN embeddings

Our fourth method consists of using the embeddings of a neural network (NN) which is trained to predict an IPA segment's immediate neighbours. As it is the case with the phon2vec model, such a network requires a training dataset of IPA sequences. During training, each IPA segment of each sequence is embedded, processed by a regular dense layer, and used to predict the segment's one-hot encoded preceding and following segments through two softmax layers. The network's architecture is depicted in Figure 2.2.

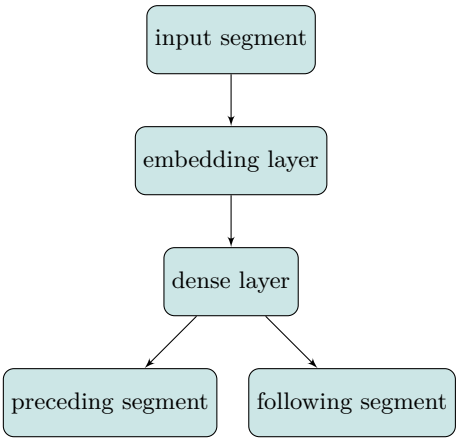

Figure 2.2: NN: layers

The network's hyperparameters include the size of the embeddings layer, the size of the dense layer, the number of training epochs, the batch size, the loss and optimisation functions. We use sparse categorical cross-entropy as the loss function and we experiment with common values of the other hyperparameters. We have also experimented with predicting a larger context by extending the number of predicted neighbours to two in each direction, but this fails to improve the resulting embeddings.

A major difference between phon2vec and the neural network, which is also one of the reasons to propose the latter, is that phon2vec does not model the order of IPA segments, effectively treating the preceding and succeeding neighbours equivalently. That is why we expect the NN embeddings to perform better than their phon2vec counterparts.

2.5 RNN embeddings

Our last method consists of using the embeddings of a recursive neural network (RNN). Given a pair of sequences, the network recursively encodes the first sequence into a vector which is then recursively decoded into an output sequence. The backward propagation of the mismatches between the output sequence and the pair's second sequence, represented as a vector of one-hot vectors, affects the whole network. Introduced by Cho et al. (2014), networks of such architecture are typically employed for machine translation, but our network is trained on pairs of IPA sequences comprising bilingual pairs of words linked to the same meaning. As both encoded and decoded data are of the same form, the embedding layer is shared. The network's architecture is depicted in Figure 2.3.

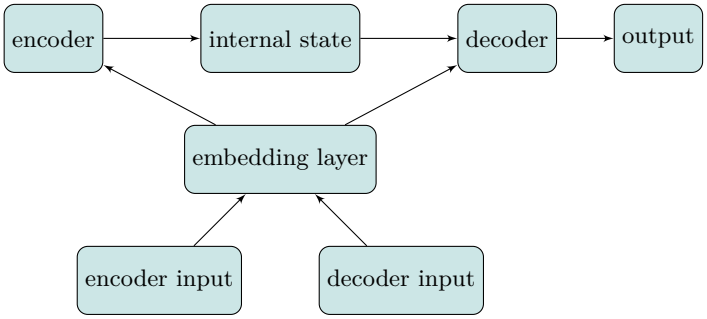

Figure 2.3: RNN: layers

The network's hyperparameters include the size of the embeddings layer, the size of the encoder and decoder recurrent layers, the number of training epochs, the batch size, the loss and optimisation functions. We use categorical cross-entropy as the loss function and we experiment with common values of the other hyperparameters. We have also experimented with using the NN embeddings as initial weights of the RNN embedding layer; this consistently improves the resulting vector representations — which we refer to as NN + RNN embeddings.

Chapter 3

Experiments

In order to evaluate the performance of the methods put forward in the last chapter, we use the Benchmark Database for Phonetic Alignments (BDPA) compiled by List and Prokić (2014). The database contains 7198 aligned pairs of IPA sequences collected from 12 source datasets, covering languages and dialects from 6 language families (cf. Table 3.1). The database also features the small set of 82 selected pairs used by Covington (1996) to evaluate his method, encoded in IPA.

	pairs	langs	family	source
Andean	619	20	Aymaran, Quechuan	Heggarty (2006)
Bai	889	17	Sino-Tibetan	Wang (2006), Allen (2007)
Bulgarian	1519	196	Indo-European	Prokić et al. (2009)
Dutch	500	62	Indo-European	Schutter et al. (2005)
French	712	62	Indo-European	Gauchat et al. (1925)
Germanic	1110	45	Indo-European	Renfrew and Heggarty (2009)
Japanese	219	10	Japonic	Shirō (1973)
Norwegian	501	51	Indo-European	Almberg and Skarbø (2011)
Ob-Ugrian	444	21	Uralic	Zhivlov (2011)
Romance	297	8	Indo-European	Renfrew and Heggarty (2009)
Sinitic	200	38	Sino-Tibetan	Hóu (2004)
Slavic	188	5	Indo-European	Derksen (2008)

Table 3.1: The BDPA datasets: numbers of pairs and languages, family affiliations as per Hammarström et al. (2018), and sources

Our methods are also compared against SCA, the current state-of-the-art method for aligning IPA sequences developed by List (2012), which is briefly described in the introductory chapter.

In order to quantify the methods' performance, we employ an intuitive evaluation scheme similar to the one used in Kondrak and Hirst (2002): if, for a given word pair, m is the number of alternative gold-standard alignments and n is the number of correctly predicted alignments, the score for that pair would be $\frac{n}{m}$. In the common word pair case of a single gold-standard alignment and a single predicted alignment, the latter would yield 1 point if it is correct and 0 points otherwise; partially correct alignment do not yield points. The percentage scores are obtained by dividing the points by the total number of pairs.

3.1 Training

Obtaining vector representations with the phon2vec and neural network methods involves settings the models' hyperparameters and training on a dataset of IPA sequences (or pairs thereof).

Our training data is sourced from NorthEuraLex, a comprehensive lexicostatistical database that provides IPA-encoded lexical data for languages of, primarily but not exclusively, Northern Eurasia (Dellert and Jäger, 2017). At the time of writing the database covers 1016 concepts from 107 languages, resulting in 121 614 IPA transcriptions. The latter are tokenised using ipatok, an open source Python package for tokenising IPA strings developed by us.[1] The phon2vec and NN embeddings are trained on the set of all tokenised transcriptions. As NorthEuraLex does not include cognacy information, the RNN embeddings are trained on the set of tokenised transcriptions of the word pairs constituting probable cognates — pairs in which the words belong to different languages, are linked to the same concept, and have normalised Levenshtein distance lower than 0.5. Admittedly, this threshold value is chosen somewhat arbitrarily; we have also experimented with thresholds of 0.4 and 0.6, but setting the cutoff at 0.5 yields better-performing embeddings.

For each of the methods, we have run the respective model with the Cartesian product of common values for each hyperparameter, practically performing a crude search of the hyperparameter space. The values we have experimented with, as well as the best-performing combinations thereof, are summarised in Tables 3.2 and 3.3.

	phon2vec	experimental values
model architecture	CBOW	CBOW, skip-gram
embedding size	15	5, 15, 30
context size	2 (one segment in each direction)	0, 2, 4
negative samples	1 per positive sample	0, 1, 2
epochs	5	5, 10, 20

Table 3.2: Hyperparameter values, phon2vec

	nn	rnn	experimental values
embedding size	64	64	32, 64
dense layer size	128	-	64, 128
rnn layer size	-	128	64, 128
epochs	10	5	5, 10, 20
batch size	128	128	32, 64, 128
optimisation	sgd	rmsprop	cf. Keras' optimisation functions

Table 3.3: Hyperparameter values, neural networks

[1] `https://pypi.python.org/pypi/ipatok`

3.2 Results

The percentage-score evaluation of the output of running our proposed methods and SCA on the BDPA datasets is summarised in Table 3.4.

	one-hot	phoible	phon2vec	nn	nn + rnn	sca
Andean	85.66	87.31	97.25	99.34	99.50	99.67
Bai	52.55	62.77	61.25	74.72	75.52	83.45
Bulgarian	60.54	80.54	77.98	82.55	86.70	89.34
Dutch	14.16	25.65	26.00	32.50	32.50	42.20
French	42.94	62.92	68.94	74.30	77.04	80.90
Germanic	39.93	51.78	54.59	71.83	72.55	83.48
Japanese	53.56	65.04	73.74	62.71	71.08	82.19
Norwegian	59.39	78.87	73.69	83.53	89.06	91.77
Ob-Ugrian	59.58	77.87	73.35	78.04	82.55	86.04
Romance	40.48	71.28	63.16	76.37	77.55	95.62
Sinitic	27.34	28.57	30.75	72.46	74.04	98.95
Slavic	76.96	90.73	84.22	89.89	96.81	94.15
Global	51.83	66.64	66.99	75.88	78.45	84.84
Covington	60.61	82.42	80.18	82.52	82.52	90.24

Table 3.4: Scores, as percentage of total alignment pairs

3.3 Discussion

The first point we would like to draw attention to is that the one-hot encoding scores are consistently lower than those in the other columns. This is expected because, unlike the other methods, one-hot encoding cannot represent variability in the degree of phonetic similarity between IPA segments. Viewing the one-hot encoding scores as a baseline, we conclude that the other methods' distance measures do indeed contribute to the task of sequence alignment.

The PHOIBLE feature vector are roughly on par with the phon2vec embeddings, yield better results than the NN embeddings on two of the datasets (Japanese and Slavic), and are outperformed by the RNN + NN embeddings. The better scores achieved by SCA and the the data-induced embeddings can be explained by the binary nature of PHOIBLE's vectors and the smaller number of features. Furthermore, PHOIBLE does not provide feature vectors for all IPA segments encountered in the BDPA datasets.

Not included in Table 3.4 are the scores of the PHOIBLE-PC method, which are instead contrasted with the scores of the base PHOIBLE method in Table 3.5. PCA-reduced vectors of size 29 yield the highest scores but still perform consistently worse than their non-reduced counterparts, albeit with relatively small difference. Thus our hypothesis that applying PCA on the PHOIBLE vector space would improve the results is proven false.

Of the data-driven methods, phon2vec yields the lowest scores, being outperformed by both neural network models in all datasets except Japanese. Given that both the phon2vec and the NN embeddings are trained on the same data, we believe that the consistent difference is due to the fact that the phon2vec model ignores the order of IPA segments.

	phoible	phoible-pc
Andean	87.31	82.31
Bai	62.77	00.23
Bulgarian	80.54	78.79
Dutch	25.65	24.53
French	62.92	61.31
Germanic	51.78	45.89
Japanese	65.04	62.56
Norwegian	78.87	75.80
Ob-Ugrian	77.87	76.20
Romance	71.28	71.04
Sinitic	28.57	00.00
Slavic	90.73	86.17

Table 3.5: Scores: PHOIBLE vs. PHOIBLE-PC

The NN + RNN model yields higher scores than the NN model that it builds upon, indicating that the embeddings of the former capture useful information that the embeddings of the feedforward network do not — the difference between the two models ranges from none for the Dutch dataset to almost 7 percent points for the Slavic dataset.

For all but the Slavic dataset, SCA yields higher scores than our best-performing NN + RNN embeddings. The score differences exhibit considerable variance — from less than 1 percent point for the Andean dataset up to 26 percent points for the Sinitic dataset. A possible explanation for this variance is the fact that not all IPA segments found in the benchmark datasets are found in the training data. For example, NorthEuraLex includes a single tonal language, Mandarin Chinese, and the NN model cannot produce meaningful embeddings for most of the tones encountered in the Sinitic and Bai datasets. Arguably, a larger training dataset featuring a richer set of IPA segments would produce better-performing embeddings.

Chapter 4

Conclusion

In this paper we have proposed, implemented, and evaluated a small set of methods for obtaining vector representations of IPA segments for the purposes of pairwise IPA sequence alignment. With the exception of a single testing dataset, our vector representations fail to outperform the current state-of-the-art IPA alignment method. Nevertheless, we consider the results of the data-driven methods not too far off the mark, and we believe that they could be significantly improved by using larger and more diverse training data. This constitutes one direction for future experiments; another possibility is to train and use embeddings specific to a particular language family or macro-area. Further investigation is also needed with respect to comparing and evaluating the methods, especially in the context of a larger application, such as cognacy identification or phylogenetic inference.

References

B. Allen. *Bai Dialect Survey*. SIL International, 2007. URL `https://www.sil.org/resources/archives/9121`.

J. Almberg and K. Skarbø. Nordavinden og sola. en norsk dialektprøvedatabase på nettet, 2011. URL `http://www.hf.ntnu.no/nos/`.

C. Brown, E. W. Holman, S. Wichmann, and V. Velupillai. Automated Classification of the World's Languages: A Description of the Method and Preliminary Results. *STUF - Language Typology and Universals Sprachtypologie und Universalienforschung*, 61, 2008. doi: 10.1524/stuf.2008.0026.

C. H. Brown, E. W. Holman, and S. Wichmann. Sound Correspondences in the World's Languages. *Language*, 89(1):4–29, 2013. ISSN 1535-0665. doi: 10.1353/lan.2013.0009.

D. Chen and C. Manning. A Fast and Accurate Dependency Parser using Neural Networks. pages 740–750. Association for Computational Linguistics, 2014.

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*, 2014.

F. Chollet et al. Keras, 2015. URL `https://github.com/keras-team/keras`.

M. A. Covington. An Algorithm to Align Words for Historical Comparison. *Comput. Linguist.*, 22(4):481–496, 1996. ISSN 0891-2017.

M. A. Covington. Alignment of Multiple Languages for Historical Comparison. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, COLING '98, pages 275–279, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics. doi: 10.3115/980451.980890.

J. Dellert and G. Jäger, editors. *NorthEuraLex (version 0.9)*. Eberhard Karls Universität Tübingen, Tübingen, 2017. URL `http://northeuralex.org`.

R. Derksen, editor. *Etymological dictionary of the Slavic inherited lexicon*. Number 4 in Leiden Indo-European Etymological Dictionary Series. Brill, Leiden and Boston, 2008.

L. Gauchat, J. Jeanjaquet, and E. Tappolet, editors. *Tableaux phonétiques des patois suisses romands*. Attinger, Neuchâtel, 1925.

H. Hammarström, S. Bank, R. Forkel, and M. Haspelmath, editors. *Glottolog 3.2.* Max Planck Institute for the Science of Human History, Jena, 2018. URL `http://glottolog.org`.

B. Hayes. *Introductory Phonology.* Blackwell, 2009.

P. Heggarty. Sounds of the andean languages, 2006. URL `http://www.quechua.org.uk`.

J. Hóu, editor. *Xiàndài Hànyǔ fāngyán yīnkù.* Shànghǎi Jiàoyù, Shànghǎi, 2004.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

G. Jäger. Phylogenetic Inference from Word Lists Using Weighted Alignment with Empirically Determined Weights. *Language Dynamics and Change*, 3 (2):245–291, 2013. ISSN 2210-5832. doi: 10.1163/22105832-13030204.

G. Jäger and P. Sofroniev. Automatic cognate classification with a support vector machine. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 128–134, 2016.

G. Kondrak. A New Algorithm for the Alignment of Phonetic Sequences. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 288–295, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

G. Kondrak. Phonetic alignment and similarity. *Computers and the Humanities*, 37(3):273–291, 2003.

G. Kondrak and G. Hirst. *Algorithms for language reconstruction*, volume 63. University of Toronto Toronto, 2002.

J.-M. List. SCA: Phonetic Alignment based on sound classes. *New Directions in Logic, Language and Computation*, pages 32–51, 2012.

J.-M. List and J. Prokić. A benchmark database of phonetic alignments in historical linguistics and dialectology. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 288–294, 2014.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, 2013.

S. R. Moisik and J. H. Esling. The 'whole larynx' approach to laryngeal features. In *Proceedings of the International Congress of Phonetic Sciences (ICPhS XVII)*, pages 1406–1409, 2011.

S. Moran, D. McCloy, and R. Wright, editors. *PHOIBLE Online.* Max Planck Institute for Evolutionary Anthropology, Leipzig, 2014. URL `http://phoible.org/`.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

C. Notredame, D. G. Higgins, and J. Heringa. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

J. Prokić, J. Nerbonne, V. Zhobov, P. Osenova, K. Simov, T. Zastrow, and E. Hinrichs. The computational analysis of bulgarian dialect pronunciation. *Serdica journal of computing*, 3(3):269–298, 2009.

C. Renfrew and P. Heggarty. Languages and origins in Europe, 2009. URL `http://www.languagesandpeoples.com`.

G. d. Schutter, B. van den Berg, T. Goeman, and T. de Jong. Morfologische atlas van de nederlandse dialecten, 2005. URL `http://www.meertens.knaw.nl/mand/database/`.

H. Shirō. Japanese dialects. *Diachronic, areal and typological linguistics*, pages 368–400, 1973.

M. P. Silfverberg, L. Mao, and M. Hulden. Sound Analogies with Phoneme Embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 136–144, 2018.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981. ISSN 0022-2836.

S. v. d. Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, 13:22–30, 2011. doi: 10.1109/MCSE.2011.37.

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

S. Wichmann, E. W. Holman, and C. H. Brown, editors. *The ASJP Database (version 17)*. 2016. URL `http://asjp.clld.org/`.

M. Zhivlov. Annotated swadesh wordlists for the ob-ugrian group (uralic family). *The Global Lexicostatistical Database*, 2011. URL `http://starling.rinet.ru/new100/main.htm`.

R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010. ELRA. URL `http://is.muni.cz/publication/884893/en`.