An Expectation Maximisation Algorithm for Automated Cognate Detection

Roddy MacSween
Homerton College & Computer Laboratory
University of Cambridge
rlm72@cantab.ac.uk

Andrew Caines **Paula Buttery**
Computer Laboratory
University of Cambridge
{apc38|pjb48}@cam.ac.uk

Abstract

In historical linguistics, cognate detection is the task of determining whether sets of words have common etymological roots. Inspired by the comparative method used by human linguists, we develop a system for automated cognate detection that frames the task as an inference problem for a general statistical model consisting of observed data (potentially cognate pairs of words), latent variables (the cognacy status of pairs) and unknown global parameters (which sounds correspond between languages). We then give a specific instance of such a model along with an expectation-maximisation algorithm to infer its parameters. We evaluate our system on a dataset of 8140 cognate sets, finding its performance of our method to be comparable to the state of the art. We additionally carry out qualitative analysis demonstrating various advantages it has over existing systems. We also suggest several ways our work could be extended within the general theoretical framework we propose.

1 Introduction

In historical linguistics, two words are deemed cognate if they share a root in a parent language, for example German 'Nacht' and English 'night' which both originate from Proto-Indo-European **nókʷts*. The task of cognate detection is interesting in its own right, but is also an important part of the larger task of proto-language reconstruction.

As described in Campbell (2013), the *comparative method* used by historical linguists to reconstruct an ancestral language from a set of potentially cognate words in daughter languages consists of three steps: assembling cognates, establishing sound correspondences and reconstructing proto-sounds. Automation of this method is made challenging by the fact that "the comparative linguist typically jumps back and forth among these steps" because of a mutual dependence between cognacy judgements and hypothesised sound correspondences. Whether two words should be deemed cognate depends on whether the sounds in them correspond according to known rules for the languages. For instance, the *t* and *ts* sounds in English and German are known to correspond, which is evidence for "tooth" and "Zahn" being cognate. But sound correspondence rules are themselves theorised based on which sounds coincide in known cognate pairs. Therefore a linguist must iteratively adapt their hypotheses about which words are cognate and which sounds correspond until they can reach a definite conclusion.

Existing approaches to automated cognate detection (ACD) fail to fully capture this idea of dealing with mutual dependence using an iterative method. Some early approaches are not iterative at all, while several more recent methods are iterative to some extent but either only carry out a small fixed number of iterations or use incomplete and ad hoc methods to update sound correspondences based on tentative cognacy judgements. In this paper we design and implement an iterative algorithm that uses the method of *expectation maximisation* for statistical inference, which is close to historical linguists' method of updating sound correspondences in one iteration based on cognacy judgements from the previous iteration.

This probabilistic approach is the main novel feature of our work, but we also build on existing approaches in other ways. Other than the work of List (2012), previous computational methods generally use little linguistic theory as a basis for making cognate judgements; in contrast, our model uses *phonological features* as a factor in determining how likely phones are to correspond. We also use a new method of clustering to turn cognate judgements between pairs of words into sets of cognates from multiple languages. We evaluate our system on a dataset consisting of 8140 cognate sets

Proceedings of the 24th Conference on Computational Natural Language Learning, pages 476–485
Online, November 19-20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

partitioned into 10 typological groups originated by List (2012).

2 Related work

Several approaches to ACD exist, with a review of the majority given in Rama et al. (2018). Most of these have the same overall structure: given a list of sets of possible cognates, they align words using some metric (typically based on the phones that make up the words) and use some function of scores of alignments to judge cognacy. The alignment process generally uses methods from bioinformatics such as the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970).

Normalised Edit Distance (Nerbonne and Heeringa, 1997) (NED) performs alignment using a manually specified distance matrix for phones. It has been used with a simple binary measure (distance only depends on whether or not two phones are the same) and more complex measures. This was a relatively early method and is not iterative.

Rama et al. (2018) discuss two methods that do not use alignment. Consonant Class Matching (CCM) (Turchin et al., 2010) is a very simple approach: it determines whether words are cognate based on whether their first consonants fall into the same sound classes. On the other hand, the system of Jäger et al. (2017) is more complex; it uses similarity as measured by other methods such as those below as a feature source for classification of cognacy using Support Vector Machines (SVMs).

The online Pointwise Mutual Information method (Rama et al., 2017) (PMI) is similar to ours in that it performs alignment with a distance matrix for pairs of phones which is adjusted iteratively. However, they do not update weights in a probabilistic way based on tentative cognacy probabilities at each iteration in the same way as us. We take into account all pairs of possible cognates in the dataset weighted by their estimated cognacy probability; in comparison they use a fixed set of cognate words deemed to be probably cognate, and these are treated uniformly. They also do not set weights in a linguistically motivated way. The system of Steiner et al. (2011) is similar to PMI, but using the LZ78 algorithm (Ziv and Lempel, 1978) to produce weights instead of PMI scores. Gilman (2012) describes another similar algorithm.

We use the LexStat algorithm (List, 2012) as a baseline in evaluation of our system. LexStat is partially iterative: it uses a simple heuristic to determine which pairs of words are definitely cognate and then sets weights for an alignment step based on those. In contrast, our system can iterate indefinitely rather than just twice. Unlike methods such as PMI, LexStat does use linguistic theory in judging cognacy of word pairs. Rather than computing distance between individual phones, it groups phones into sound classes (by place and manner of articulation) and then considers combinations of sound classes and prosodic contexts as the segments used for alignment. The use of sound classes is similar to our framework with phonological features, although ours is in some ways more general. Prosodic context is not taken into account by our system and could be a future extension.

Overall, the primary difference between our system and previous work is the way in which we adjust parameters for how likely two sounds are to correspond based on intermediate estimates of cognacy probabilities. Producing theories about which sounds correspond based on cognacy judgements is a key part of the method followed by human linguists, but many existing automated methods do not do this at all. Those that do often only make a limited number of adjustments, or lack theoretical justification for the approach used. In contrast, in our system this process has equal status to the inference step in the opposite direction, and has a rigorous probabilistic interpretation within the expectation-maximisation framework. The probabilistic framework used also has the benefit of allowing easier identification of limitations of the system, such as cases where unrealistic assumptions of independence are made. A key element of our system that enables this approach is the fact that we separately model the processes of generating two cognate words from a common ancestor and two non-cognates from different ancestors. In comparison, previous work implicitly models only the former.

2.1 PanPhon

We initialise our parameters using weights derived from the PanPhon database (Mortensen et al., 2016) of phonological features with the standard categorisation of Chomsky and Halle (1968). These are binary or ternary variables representing possible axes of variation of phones. For example, the [+/-voice] feature distinguishes between sounds such as *b*, *d* and *z* where the vocal folds vibrate during articulation and those such as *p*, *t* and *k* where they do not. This is relevant to our cognate detection because phones with similar feature sets are more likely to correspond in cognate pairs. Not all features have equal status for this, for instance the [+/-syllabic] feature is especially useful.

Human historical linguists form theories about evolution of languages involving complex relationships between multiple features (for example, the change of certain stops from voiced to voiceless between Proto-Indo-European and Proto-Germanic). Our system currently uses phonological features in a simple language-independent way, however one advantage it has is the fact that these sophisticated linguistic theories could in principle be integrated into it simply by altering the initialisation code, without changing the overall framework. This lowers the barrier for linguists to experiment with adding their domain knowledge into an automated cognate detection system, since a wide variety of linguistic models could be implemented with changes to only a small section of the codebase.

3 Model

3.1 General framework

We can formulate cognate detection as a probabilistic inference task, where for a dataset of n pairs of words we have a sequence $\boldsymbol{X}$ of n random variables representing the generation of the pairs, a sequence $\boldsymbol{Z}$ of n indicator variables for the event of each pair being cognate, and some global parameters $\boldsymbol{\theta}$ that $P(\boldsymbol{X}, \boldsymbol{Z})$ depends on. Then cognate detection can be done by finding values for $\boldsymbol{\theta}$ maximising the likelihood of the observed words

$$\sum_{\boldsymbol{z} \in \{0,1\}^n} P(\boldsymbol{X}, \boldsymbol{Z} = \boldsymbol{z} \mid \boldsymbol{\theta}) \tag{1}$$

then using the probability distribution these give over $\boldsymbol{Z}$ to judge cognacy.

It is difficult to do this maximisation directly. Instead, we can use an expectation-maximisation (EM) algorithm (Dempster et al., 1977):

- Initialise $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$
- Find the distribution of $\boldsymbol{Z}$ for $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$ (expectation step)
- Update $\boldsymbol{\theta}^{(t+1)}$ to take the values maximising the expectation with respect to the above distribution of the log likelihood of the observed data (maximisation step)
- Repeat expectation and maximisation steps until $\boldsymbol{\theta}^{(t)}$ converges

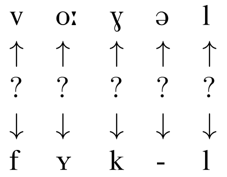

Figure 1: Generation of *voːɣəl* and *fʏkl* (Dutch and Icelandic words for ‘bird’) as cognates

3.2 Cognacy model

3.2.1 Definition

A model must be chosen that makes the maximisation step computationally feasible. Here, we use two separate models for the case where a pair of words are cognate and the case where they are not. For two words w and w' we model $P(X_i = (w, w') \mid Z_i = 1)$ by assuming the words have been generated on a phone-by-phone basis from some word in the parent language as shown in Figure 1. Each pair of arrows is associated with a parameter, for instance θ_{vf} is the probability that a random phone in the parent language would generate *v* in Dutch and *f* in Icelandic. Then we have

$$P(X_n = (w, w') \mid Z_n = 1) = \prod_i \theta_{\texttt{align}(w,w')_i} \tag{2}$$

where $\texttt{align}(w, w')$ is the sequence of pairs of aligned phones in the two words,

Figure 2 shows the similar model for $P(X_i = (w, w') \mid Z_i = 0)$. Here each individual arrow is associated with a parameter α_x. Formally we have

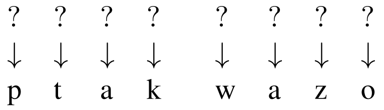

Figure 2: Generation of *ptak* and *wazo* (Polish and French words for “bird”) from separate etymons

$$P(X_n = (w, w') \mid Z_n = 0) = \prod_i \alpha_{\texttt{first}(\texttt{align}(w,w')_i)} \prod_j \alpha_{\texttt{second}(\texttt{align}(w,w')_j)} \tag{3}$$

where `first` and `second` are the first and second components of pairs of the alignment of the words. We use alignments rather than the words themselves even in the non-cognate case[1] because if we used the words directly this would imply a model where deletions only occur for words which are cognate, which is not realistic.

4 Algorithm

4.1 E-step

The expectation step of our algorithm involves for each pair of words $X_n = (w, w')$ computing the probability under some set of parameters $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$, $\boldsymbol{\alpha'}$ that they are cognate. By Bayes' rule we have

$$P(Z_n = c \mid X_n = (w, w')) = \frac{P(X_n = (w, w') \mid Z_n = c)P(Z_n = 1)}{\begin{array}{c}P(X_n = (w, w') \mid Z_n = 0)P(Z_n = 0) + \\ P(X_n = (w, w') \mid Z_n = 1)P(Z_n = 1)\end{array}} \tag{4}$$

We compute $P(X_n = (w, w') \mid Z_n = 0)$ and $P(X_n = (w, w') \mid Z_n = 1)$ using Equation 2 and Equation 3. $P(\text{cognate})$ and $P(\neg\text{cognate})$ can be either fixed values, or vary during the iteration.

Calculating $P(X_n = (w, w') \mid Z_n)$ requires an alignment of w and w'. We produce a new alignment at each iteration by using the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) with the logarithms of $\boldsymbol{\theta}$ values as weights. This gives an alignment with the maximum likelihood of cognacy according to the model, since the Needleman-Wunsch algorithm involves summing weights (which is equivalent to multiplying the original probabilities). If there are multiple best alignments, one is chosen arbitrarily.

4.1.1 M-step

The maximisation step takes probabilities of cognacy for all pairs of words and infers the best fitting parameters. For the $\boldsymbol{\alpha}$ parameters the process for this is analogous to simple maximum likelihood estimation with known values for $\boldsymbol{Z}$. In that case, for a phone x we would estimate α_x as the frequency of x across all words in non-cognate pairs in the relevant taxon. Here we instead have a distribution over $\boldsymbol{Z}$ so we use the expected frequency:

$$\alpha_x = \frac{E(\#x)}{\sum_y E(\#y)} \tag{5}$$

where $E(\#x)$ is the number of occurrences of phone x in words of the relevant taxon, with the count for each occurrence weighted by the probability that the containing word is not part of a cognate pair.

The equivalent formula could be used to estimate θ parameters, however this might cause problems because some pairs of corresponding phones may not occur in the dataset but still be modelled as having non-zero probability.

To deal with this, we use *additive smoothing* (Manning et al., 2008) for estimation of θ parameters. In general, given a random variable X with d discrete values and n trials, a smoothed estimator for each parameter of X is

$$P(X = x) = \frac{\#(X = x) + \beta}{n + \beta d} \tag{6}$$

where $\#(X = x)$ is the number of times x occurred and β is a smoothing parameter.

Additive smoothing typically uses the same β value for each parameter. This is appropriate when there is no relevant prior information. However, in this case there is some information that should be taken into account: frequent pairs are on average made up of frequent individual phones. Therefore rather than using a constant smoothing parameter, we weight it for each pair by the frequencies of its component phones, giving an estimation of

$$\theta_{xy} = \frac{E(\#(x, y)) + \beta\frac{E(\#x)E(\#y)}{\#\text{pairs}}}{\sum_{(z,w)} E(\#(z, w)) + \beta\#\text{pairs}} \tag{7}$$

where $E(\#(x, y))$ is the count of alignments of phones x and y weighted by probabilities of containing words being cognate, $E(\#x)$ is the same for individual phones, and $\#$pairs is the total number of possible pairs of phones (the product of the number of phones for each taxon). Note that $E(\#x)$ here is weighted by the probability of cognacy whereas in Equation 5 it was weighted by probability of *non*-cognacy.

4.1.2 Initialisation

Expectation maximisation is guaranteed to find a locally optimal set of parameters, however depending on the initial values they may not be globally optimal. Often this is dealt with by running the algorithm multiple times from a variety of initial parameters. However, this will not work in our case as our algorithm is very general; it can infer completely arbitrary phone-level relationships between taxons and therefore most possible sets of initial parameters will lead to poor local optima.

[1] The difference being that the aligned words may contain -, indicating insertion/deletion

Therefore we must initialise the parameters in a linguistically motivated way.

The α parameters are straightforward to set using the proportions of phones in the whole dataset (i.e. in the same way as in the maximisation step but taking the whole dataset to be noncognate).

The $\boldsymbol{\theta}$ parameters must be initialised in a more complex way. Each pair of phones is assigned a weight based on the category it falls into. These categories depend on phonological features of phones, as given by PanPhon. The categories we use are as follows:

1. Two identical phones
2. Two phones that differ in one phonological feature, for instance *t* and *d*
3. Two phones that differ in two features
4. Two phones that differ in more than two features, but have the same value for the syllabic feature (are either both consonants or both vowels[2])
5. A phone and a gap (representing a deletion)
6. All other pairs

Then weights for the pairs are normalised to give valid probabilities.

4.1.3 Clustering

Using the expectation-maximisation algorithm described above gives probabilities of cognacy for one pair of taxons, but the dataset we use has multiple taxons. Therefore some method is needed to turn scored pairs of words into clusters of cognates. Previous approaches use clustering algorithms from bioinformatics such as UPGMA (Sokal, 1958) for this (e.g. (List, 2012)) which produce clusters of cognates with low *average* distances.

We use a method that is instead concerned with the *maximum* distance between pairs in a cluster. This seems more similar to the approach a human linguist would take: when trying to determine whether a language is part of some family, they would not consider on average how close it is to each member, but rather try to find a single member that is inarguably related to it.

Our method is to create a graph where nodes are words and there is an edge between two words if their computed cognacy probability exceeds some threshold. Then the clusters of cognates are the connected components of this graph. We find these clusters using the well-known algorithm described in Hopcroft and Tarjan (1973).

5 Evaluation

5.1 Dataset

To evaluate our system we use a dataset compiled by List (2012) for use with LexStat. The dataset consists of several partitions, containing words from different taxon families. Each partition contains a list of words, with their corresponding taxons, glosses (meanings), IPA transcriptions and gold-standard cognate-set annotations. Two words are considered to be potentially cognate if they have the same gloss (and do not belong to the same taxon). Table 1 gives details about the number of glosses, cognate sets and words in each partition of the dataset we use. The original dataset has two additional partitions, but these did not have transcriptions in a suitable form and therefore were not used. The BAI partition contains tone numerals, but our system is unable to use the information they provide so they are discarded in the preprocessing step. Other than this we follow the same procedure for all partitions. Further details about the dataset are given in Section 4.3.4 of List (2012).

5.2 Method

The source code for our system is available at `https://github.com/roddyyaga/cognates/` and can be used to reproduce these results.

Our system has several hyperparameters that must be set: the weights used to produce initial $\boldsymbol{\theta}$ values, the smoothing parameter β, the number of iterations, and the threshold used in clustering. We set these by testing on the PIE section of the dataset. This did not cause overfitting, since the system does not perform better either in absolute terms or relative to other sections on that subset.

For the weights for the initial θ values, we used the following values: 25,000, 5000, 50, 20, 10, 1 for the categories in subsubsection 4.1.2.

For the smoothing parameter β we used 0.0001. We found that the sections of this dataset each contained enough datapoints that smoothing did not improve classification performance. However, a small level of smoothing did make the system more efficient, since with zero smoothing there were some cases where all alignments of a pair of words had

[2] This feature specifies whether a phone can be the nucleus of a syllable (Chomsky and Halle, 1968, 354). Vowels are [+syllabic] and (unless they are syllabic) consonants are [−syllabic].

Partition	Description	Glosses	Cognate sets	Words
BAI	Bai dialects	110	205	1028
IEL	Indo-European languages	207	1778	4393
JAP	Japanese dialects	200	458	1985
OUG	Uralic languages	110	239	2055
PAN	Austronesian languages	210	2730	4358
GER	Germanic languages and dialects	110	182	814
KSL	various languages	200	1179	1400
PIE	Indo-European languages	110	615	2172
ROM	Romance languages	110	177	589
SLV	Slavic languages	110	165	454

Table 1: Overview of the dataset used

zero probability, meaning there was a large number of highest scoring alignments which took significant time to iterate through.

For the threshold value we used 0.1. However, for most sections of the dataset using these hyperparameters the estimated cognacy probabilities for most pairs of words became very close to 0 or 1 after several iterations, therefore many threshold values would have given similar performance. Compared with an even spread of output probabilities across the range $(0, 1)$, this bimodal distribution has the drawback that it prevents interpretation of the probabilities as degrees of certainty. However, it has the advantage that it makes performance relatively independent of the choice of threshold.

Additionally, our system has a baseline probability of cognacy that must be set for each pair of taxon. We evaluated the system both using a fixed baseline (set to 0.005) and using a dynamic baseline where the value for each pair of taxons was updated after each iteration. Updates were made using the estimated cognacy probabilities for each pair of words produced in the expectation step. For each pair of taxons, we estimated a new baseline probability from these probabilities by taking the arithmetic mean across all word pairs for that taxon pair. The principle behind this dynamic adjustment is that the system can learn to behave differently for pairs of related taxons and unrelated pairs, based on how related the words in each pair appear to be at each iteration. We used a value of 0.005 as an initial baseline probability for all taxon pairs. The number of iterations was five for the fixed baseline and four for the method with updates.

Subset	LexStat	Fixed	Dynamic
PIE	**0.83**	**0.83**	**0.83**
SLV	0.94	**0.96**	**0.96**
OUG	0.92	**0.94**	**0.94**
GER	0.94	0.95	**0.96**
JAP	**0.93**	**0.93**	**0.93**
ROM	**0.94**	0.90	0.90
BAI	**0.89**	0.88	**0.89**
KSL	**0.94**	0.90	0.93
IEL	**0.81**	0.77	**0.81**
PAN	0.81	0.80	**0.83**

Table 2: Comparison of F-scores achieved by LexStat and this system (using both a fixed and dynamic baseline).

5.3 Results

We evaluated the cognacy judgements of our system using *B-cubed F-score* (List, 2012) with the LingPy library (List et al., 2019). These results are summarised in Table 2.

Overall our system performs very similarly to LexStat, and our results did not differ by a large amount depending on the method for setting the baseline. The results in List (2012) show LexStat outperforms previous systems on this dataset, and the more recent work of Rama et al. (2018) confirms that LexStat is still the best performing system (at least for the language families under consideration here). Therefore we can conclude our system performs approximately as well as the current state of the art.

The results of our system vary between partitions in a way that is not illustrated by the summary in Table 2. Figure 3 shows the F-score achieved at each iteration by our system (using the dynamic baseline). For some partitions of the dataset, the F-score

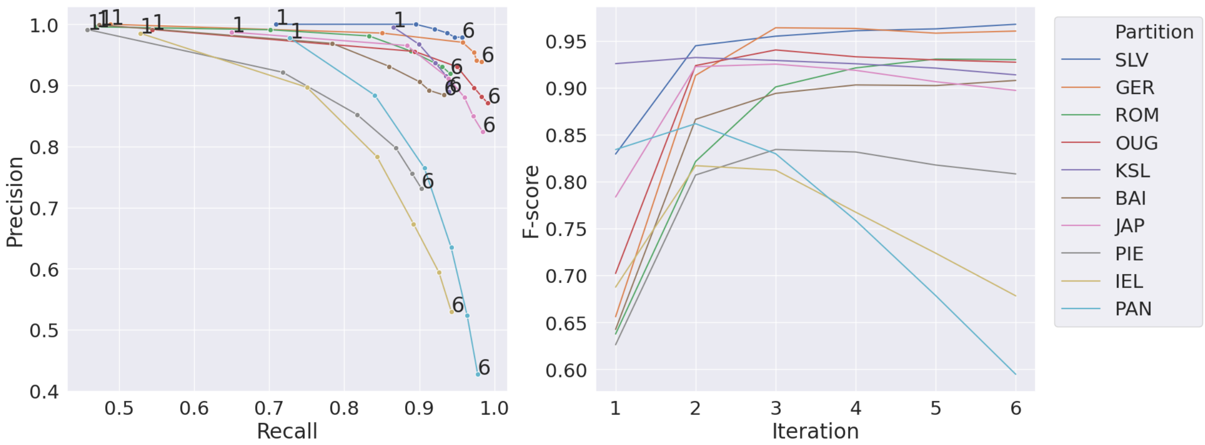

Figure 3: B-cubed precision and recall at each iteration, and B-cubed F-score at each iteration

converges and plateaus after the first few iterations, or continues to increase at every iteration. But for other partitions such as IEL and PAN it peaks after a couple of iterations and then decreases.

The root cause of this behaviour is unclear; IEL and PIE have very different results after the third iteration even though they behave similarly up to that point and cover the same language families. However from Figure 3 we can see that the cause is not that the system get worse overall at judging cognacy for the IEL and PAN partitions but rather that it is trading decreased precision for increased recall, because the distribution of estimated cognate probabilities is shifting relative to the threshold.

5.4 Error analysis

We also perform a qualitative analysis of some individual errors made by our system and LexStat on the PIE partition, and patterns among these errors.

In general, both systems tend to make the same kinds of errors. They both judge pairs of cognates with dissimilar forms as noncognate, for instance French *wazo* and Italian *utʃːɛllo* (‘bird’). Likewise, they incorrectly judge noncognate pairs that are coincidentally similar, such as Bulgarian *kutʃɛ* and Hindi *kʊttaː* (‘dog’). In these cases, it seems likely that a human without prior knowledge of historical sound changes in the relevant languages would make the same errors. In other cases, both systems make errors that a human (even without linguistic knowledge) would be unlikely to make. For example, they both judge Bulgarian *vsitʃki* and Czech *fʃɪxnʲɪ* (‘all’) as noncognate, when intuitively it seems clear that those words plus Polish *fʃistsɨ* and Russian *fsʲe* form a cognate set.

The converse tendency – incorrectly grouping together sets of words that intuitively appear noncognate – is only observed in our system and not LexStat. For instance, our system groups the Romance words for “all” (French *tu*, Portuguese *todu* etc.) with the Germanic words (Dutch *alə*, English *ɔːl* etc.) while LexStat correctly distinguishes the two. However, weaker relative performance in this regard by our system is compensated for by better performance for glosses where there are only one or two cognate sets. For example, LexStat produces 5 cognate sets for the word “salt”: one each for the families of Slavic, Germanic and Romance languages (except French), one with Armenian and French, and one with just Greek. In comparison, apart from Armenian (which is incorrectly put in its own set) our system correctly reaches the natural conclusion[3] that all the words are cognate.

While LexStat does not incorrectly merge large distinct cognate sets, both systems do incorrectly judge individual noncognate pairs even when they have dissimilar forms. This most frequently occurs for short words, where chance resemblance between one or two pairs of phones has a large overall effect (for instance *f* and *v* in Greek *fiði* and Polish *võ ʒ* (‘snake’)).

One interesting error that reveals a shortcoming of our system’s clustering method is its judgement of Italian *vɛntre* as cognate with Italian *textipapantʃa*. By definition it is not possible for two words from the same taxon to be cognate with itself, and our system will not judge them as such directly. But it is possible for the clustering algorithm to produce these judgements indirectly since cognacy is transitive. In this case, one word in

[3] Given their phonetic similarity, e.g. Bulgarian *sol*, English *sɔːlt*, French *sɛl* and Italian *sale*

the set of French *vɑ̃ tʁ*, Portuguese *vẽtɾə*, Spanish *bjentre* and Italian *vɛntre* has been (incorrectly) judged as cognate with one of Italian *pantʃa* and Romanian *pəntetse*. However, it is not obvious how to resolve this, since some alternative clusterings that avoid this impossibility (for instance in this case putting *vɛntre* in its own cognate set) would be strictly worse. One possible approach might be considering all sets of edges in the cognacy probability graph where removing them would give a possible clustering and removing the set with the lowest total probability.

There are several patterns in the "obvious" errors produced by both systems. Firstly, relationships between more than two words with the same gloss are often ignored. For instance, the errors described above with *vsitʃki* and cognates could be avoided by taking into account the fact that those words form part of a mutually similar set rather than considering their relationships with other words in that set in isolation. Similarly, the fact that the taxons under consideration form families is ignored. For example, there are several cases where the words from Romance languages form one cognate set, but one or both systems group them as two sets: one with the French word by itself and another containing all the others. Finally, both systems treat the sequences of phones that form words uniformly. But there are many cases where cognate pairs of words are more similar at the start than the end, and in particular where one word is shorter than the other. Several examples of this occur for French, for instance *sɛ̃* is cognate with Italian *seno* ('blood'). One way to adapt our cognacy model to deal with this would be placing a greater emphasis on phone correspondence probabilities at the start of the word.

5.5 Analysis of sound correspondence parameters

We can examine the behaviour of the system by looking at how the $\boldsymbol{\theta}$ parameters change at each iteration. For instance, after the first iteration on the PIE subset the 15 pairs with the highest probabilities for English and German are (-, ə), (*s*, *z*), (ə, ə), (*a*, *a*), (*t*, *t*), (*b*, *b*), (*s*, *s*), (*ɪ*, *ɪ*), (*h*, *h*), (*æ*, *a*), (*f*, *f*), (*l*, *l*), (*d*, *t*), (*r*, *r*) and (*n*, *n*) – predominantly pairs of identical phones. This is unsurprising given the high initial weights given to these.

After five iterations the pairs with the probabilities are quite different, the top 15 being (*t*, *t*), (*s*, *z*), (*t*, *ts*), (*h*, *h*), (*ɪ*, *ɪ*), (*t*, *s*), (*ʋ*, *v*), (*f*, *f*), (*m*, *m*), (*l*, *l*), (*d*, *t*), (-, *n*), (-, ə), (*n*, *n*) and (*r*, *r*). Many of these such as (*ʋ*, *v*) and (*t*, *ts*) are well-known sound correspondences, validating our method. Similar results are observed for other pairs of taxons.

5.6 Armenian-Greek case study

We also tested our system on a subset of the PIE partition containing only Armenian and Greek words. Many of the sound changes between Proto-Indo-European and Armenian are unusually dramatic, for instance the change from **dw* to *erk*. This makes detecting cognates between these languages considerably more challenging task. We also use this case study to demonstrate how initialisation of the model's parameters can be done in in linguistically motivated and language specific ways.

This new dataset consists of 24 cognate pairs of words and 75 noncognate pairs. It was produced by dropping words from taxons other than Armenian and Greek from the PIE partition, and also dropping words from those taxons when only one taxon had words for a gloss.

This dataset contains much smaller cognate sets than the overall dataset (each has either 1 or 2 elements) and is skewed towards noncognate pairs, while our focus in this analysis is more on achieving correct detection of cognates rather than correctly avoiding false claims of cognacy. These factors make B-cubed F-score an unsuitable metric to use, as high scores for this can be achieved by judging all pairs of words as noncognate. Therefore instead we consider accuracy among cognate and noncognate pairs separately.

As a baseline, we evaluated LexStat on this dataset. We found that regardless of the threshold used, it judged all pairs of words as noncognate, since there were no obvious patterns of sound correspondence.

We then tested our system using three methods for setting initial parameters. First we used the same weights as before ("original weights"). Second, we used that method except with the weights for identical phones being reduced from 25,000 to 8500 ("lowered identical weights"). Third, we used a variation of the second method where the weights for several individual pairs of phones increased to 8500 ("increased reflex weights"). Proto-Indo-European **d* has reflexes *ð* and *t* in Greek and Armenian respectively. Similarly, the consonant cluster **dw* has reflex *ð* in Greek and *ɛrk* in Arme-

Method	Cog.	Noncog.	Overall
LexStat	0.00	**1.00**	**0.75**
Original	0.13	0.96	**0.75**
Lowered identical	0.29	0.83	0.69
Increased reflex	**0.54**	0.83	**0.75**

Table 3: Accuracies for all pairs plus cognate and noncognate subsets on the Armenian-Greek data. Results for our system were produced after 3 iterations.

nian. Reflecting these correspondences, we set the weights for the pairs (*ð*, *t*), (*ɛ*, -), (*ɾ*, *ð*) and (*k*, -) to 8500.

We experimented with increasing the weights for corresponding reflexes with a weight of 25,000 for identical phones, but found this gave the same results as just using the original weights. The reason for this is that assigning such a high weight to identical phones causes the initial probabilities for other categories of correspondence to be very low, and so only words with a high proportion of exactly matching phones will be given a high probability of cognacy. This is not an issue when the system is evaluated on the original partitions of the dataset, as these have many cognate pairs that are almost identical. But the Armenian and Greek words generally differ more, and in particular the pairs with the corresponding reflexes mentioned above do not contain matching identical phones. Therefore it is necessary to relax the emphasis on identical phones in order to judge these words as cognate.

Table 3 shows the accuracies produced by LexStat and these three methods. LexStat judges all pairs as noncognate, which achieves a relatively high overall accuracy since the dataset is skewed towards noncognates. In comparison, our system does judge a minority of pairs as cognate, achieving positive accuracy for cognate pairs at the cost of reduced accuracy for noncognates. The combined effect for our "original weights" and "increased reflex weights" parameter settings is an overall accuracy that is marginally higher than that of LexStat, but the more meaningful advantage of our system for challenging data such as this is that its parameters can be varied to allow cognates to be detected at all. Increasing the weights for corresponding reflexes caused a significant[4] increase in accuracy for cognate pairs while leaving accuracy for noncognate pairs unchanged. This increase comes from pairs of words containing the sound correspondences mentioned such as Armenian *tal* and Greek *ðino* ('to give') and Armenian *jɛɾku* and Greek *ðjo* ('two') being correctly judged as cognate when they were not previously. This positive effect from setting initial weights based on known linguistic relationships between languages demonstrates how our system could be used for joint human-computer cognate detection.

6 Conclusions

In this paper, we formalised the task of cognate detection as inference in a general statistical model, defined a specific example of such a model, and then designed and implemented an expectation-maximisation algorithm for that model. We evaluated its performance on an existing dataset, finding it to be comparable with the current state of the art. We also evaluated it qualitatively to demonstrate advantages it has over existing systems, such as greater flexibility on challenging datasets.

There are many ways this system could be extended by future work. The word-level model of cognacy could be made more sophisticated in various regards; one that appears especially promising from our qualitative evaluation would be to give different weights to pairs of aligned phones depending on the position they come in the word or on neighbouring phones, which would improve the system's ability to capture conditioned sound changes (Campbell, 2013, 15). A larger change would be to modify the system to perform cognate detection and phylogenetic reconstruction of taxon families simultaneously. This could significantly increase the performance of the system, since many of the errors observed here could be avoided by using information about taxon families. It would also build on our approach of mimicking the method of a human linguist by going from iteratively alternating between making cognate judgements and determining sound correspondences to iteratively alternating between those steps and reconstruction of language family trees.

Acknowledgements

This second and third authors are supported by Cambridge Assessment, University of Cambridge. We thank the anonymous reviewers for their helpful feedback.

[4] One-sided t-test, $p < 0.05$

References

Lyle Campbell. 2013. *Historical Linguistics: An Introduction*, 3rd edition. Edinburgh University Press.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row New York.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Sophia Gilman. 2012. Comparative method algorithm. *Cambridge Occasional Papers in Linguistics*, 6.

John Hopcroft and Robert Tarjan. 1973. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378.

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: Volume 1, Long Papers*, pages 1205–1216.

Johann-Mattis List. 2012. Lexstat: Automatic detection of cognates in multilingual wordlists. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, EACL 2012, pages 117–125, Stroudsburg, PA, USA. Association for Computational Linguistics.

Johann-Mattis List, Simon J Greenhill, Tiago Tresoldi, and Robert Forkel. 2019. LingPy. A Python library for quantitative tasks in historical linguistics.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

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(3):443–453.

John Nerbonne and Wilbert Heeringa. 1997. Measuring dialect distance phonetically. In *Computational Phonology: Third Meeting of the ACL Special Interest Group in Computational Phonology*.

Taraka Rama, Johann-Mattis List, Johannes Wahle, and Gerhard Jäger. 2018. Are automatic methods for cognate detection good enough for phylogenetic reconstruction in historical linguistics? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 393–400, New Orleans, Louisiana. Association for Computational Linguistics.

Taraka Rama, Johannes Wahle, Pavel Sofroniev, and Gerhard Jäger. 2017. Fast and unsupervised methods for multilingual cognate clustering. *CoRR*, abs/1702.04938.

Robert R Sokal. 1958. A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38:1409–1438.

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

Peter Turchin, Ilia Peiros, and Murray Gell-Mann. 2010. Analyzing genetic connections between languages by matching consonant classes. *Journal of Language Relationship*, 5:117–126.

Jacob Ziv and Abraham Lempel. 1978. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536.