Similarity dependent Chinese Restaurant Process for Cognate Identification in Multilingual Wordlists

Taraka Rama
Department of Informatics
University of Oslo, Norway
tarakark@ifi.uio.no

Abstract

We present and evaluate two similarity dependent Chinese Restaurant Process (sd-CRP) algorithms at the task of automated cognate detection. The sd-CRP clustering algorithms do not require any predefined threshold for detecting cognate sets in a multilingual word list. We evaluate the performance of the algorithms on six language families (more than 750 languages) and find that both the sd-CRP variants performs as well as InfoMap and better than UPGMA at the task of inferring cognate clusters. The algorithms presented in this paper are family agnostic and can be applied to any linguistically under-studied language family.

1 Introduction

Cognates are related words across languages that have descended from a common ancestral language. Identification of cognates is an important step in historical linguistics while establishing genetic relations between languages that are hypothesized to have descended from a single language that existed in the past. For instance, English *hound* and German *Hund* "dog" are cognates that go back to the Proto-Germanic stage. Cognate identification requires great amount of scholarly effort and is available for some language families such as Indo-European, Dravidian, Austronesian, and Uralic which have a long tradition of comparative linguistic research that involves decades (Dravidian family) to centuries (Indo-European family) of scholarly effort. Automatic detection of cognates with high accuracy is very much desired for reducing the effort required in analyzing understudied language families of the world.

Typically, expert annotated cognate sets are employed to infer phylogenetic trees showing language relationships that can be used to test hypotheses about temporal and spatial evolution of language families (Bouckaert et al., 2012; Chang et al., 2015), linguistic reconstruction of ancestral states on a tree (Jäger and List, 2017), or lexical reconstruction (Bouchard-Côté et al., 2013). Rama et al. (2018) showed that cognates inferred from automated methods of cognate detection can be used to infer high quality phylogenetic trees. The authors noted that there is a need for more research towards developing highly accurate cognate identification methods that can be applied to the data of not so well-studied language families which will be of assistance to historical linguists to automate parts if not the whole of the comparative method.

The last decades have seen a large amount of computational effort towards automatizing the process of cognate identification since the work of Covington (1996) and Kondrak (2002). The computational effort involved devising new sequence alignment algorithms (Kondrak, 2005, 2009), novel sound transition matrices which are linguistically guided (Kondrak, 2001; List, 2012b) or data-driven (Jäger, 2013; Rama et al., 2013, 2017; List, 2012a), and machine learning approaches (Hauer and Kondrak, 2011; Rama, 2015, 2016; Jäger et al., 2017) to identify cognates within multilingual word lists (see table 1; Swadesh, 1952) belonging to different language families and dictionaries (St Arnaud et al., 2017).

Most of the above cognate identification methods involve a workflow consisting of computation of distances between all the word pairs that have the same meaning using a machine learning algorithm or a sequence alignment algorithm; and, then clustering the pairwise distance matrix using a clustering algorithm such as InfoMap (Rosvall and Bergstrom, 2008) or UPGMA (Unweighted Pair Group Method with Arithmetic Mean; Sokal and Michener, 1958).

Both InfoMap and UPGMA require a predefined threshold that is either set heuristically or through tuned to obtain to obtain optimal perfor-

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 271–281
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics

Language	ALL	AND	...
English	ol^1	End^1	...
German	al3^1	unt^1	...
French	tu^2	e^2	...
Spanish	to8o^2	i^2	...
Swedish	ala^1	ok^3	...

Table 1: Excerpt of the Indo-European word list (from our dataset) in ASJP code for five languages belonging to Germanic (English, German, and Swedish) and Romance (Spanish and French) subfamilies. Cognates are indicated with the same superscript.

mance at identifying cognate clusters on a held-out expert annotated cognate dataset(s). The clustering threshold is a single number that is tuned for all the meanings and not separately for each of the meanings. A single global threshold can lead to poor performance since the number of cognate sets vary a lot across meanings for different language families. For instance, the Indo-European dataset has cognate cluster sizes ranging from 37 for meaning *because* to 1 for meaning *name*.

On the other hand, a non-parametric clustering method such as Chinese Restaurant Process (CRP; Gershman and Blei 2012) can form clusters directly from the data without the need for tuning the threshold. CRP has found application in different NLP tasks such as morphological segmentation (Goldwater et al., 2006), language modeling (Goldwater et al., 2011), machine translation (Ravi and Knight, 2011), part-of-speech induction (Blunsom and Cohn, 2011; Sirts et al., 2014), and language decipherment (Snyder et al., 2010).

In this paper, we present two clustering algorithms inspired from similarity dependent Chinese Restaurant Process for the purpose of inferring cognate clusters. Our CRP based clustering algorithms take a word pair similarity matrix as input and infer cognate clusters automatically without needing any threshold. The sd-CRP algorithms have a hyperparameter α that allows us to form new clusters. We compare the performance of the CRP algorithms on six different language families and find that the CRP algorithms better than UPGMA and yields better or competing performance against InfoMap. We sample α so that the algorithms are robust to the initial value of α.

The paper is organized as follows. We describe related work in section 2. In section 3, we describe the word similarity features used to train the SVM model. We describe sd-CRP, UPGMA, and InfoMap algorithms in section 4. We describe the evaluation metrics and datasets in section 5. We present the results of our experiments in section 6. We discuss the results by analyzing the effect of features on SVM model, initial α values, and missing data on the performance of clustering in section 7. Finally, we conclude and present directions for future work in section 8.

2 Related work

Most of the automated cognate identification work mentioned in the previous section employed either UPGMA or InfoMap algorithms. Hauer and Kondrak (2011) were the first to apply UPGMA clustering algorithm to infer cognate sets from Swadesh lists. The authors trained a SVM classifier based on string similarity features to calculate word distances between all word pairs for a meaning. The pair-wise distance matrix is supplied to UPGMA with a predefined threshold for inferring word clusters. The UPGMA algorithm is simple and yields reasonable results across various language families (List, 2012a). However, UPGMA clustering algorithm is dependent on the threshold that needs to be tuned to obtain optimal performance (List et al., 2017b).

The cognate identification work of Hall and Klein (2011) and Bouchard-Côté et al. (2013) requires the phylogenetic tree of the language family to be known beforehand which is an unrealistic assumption for large number of world's language families. In another work, List et al. (2016) employ a weighted variant of Levenshtein distance known as SCA (see section 3) for calculating similarity between two words. Then, they apply a community detection algorithm known as InfoMap for the purpose of discovering partial cognate sets in multiple groups of Sino-Tibetan language family. The authors find that the InfoMap algorithm works better than UPGMA when tuned for threshold. In this paper, we compare the CRP clustering algorithms against InfoMap and the similarity variant of UPGMA algorithm described in section 4.3.

3 Word similarity model

In this section, we present the word similarity features used to train our SVM model at the binary task of classifying if a word pair is cognate or non-cognate.

String similarity features We use length normalized edit distance, number of common bigrams, common prefix length, individual word lengths, and absolute difference between the word lengths as features for training a SVM classifier (Hauer and Kondrak, 2011). We refer to this feature set as HK.

Point-wise Mutual Information (PMI) We include PMI weighted Needleman-Wunsch (Needleman and Wunsch, 1970) word similarity score (Jäger, 2013) as an additional feature for training the SVM classifier. The (unweighted or vanilla) Needleman-Wunsch algorithm is the similarity counterpart of the Levenshtein distance. The vanilla Needleman-Wunsch algorithm assigns equal negative weight to a common sound correspondence such as /s/ $\sim$ /h/ and a highly improbable sound correspondence such as /p/ $\sim$ /r/. The PMI weighted sound pair matrix inferred in Jäger (2013) assigns a positive weight to common sound correspondences and a negative weight to the latter ones. The PMI weight for two sounds i and j is defined as $\log \frac{p(i,j)}{q(i) \cdot q(j)}$ where, $p(i,j)$ is the relative frequency of i, j occurring at the same position in the aligned word pairs and $q(.)$ is the relative frequency of a sound in the whole word list. The similarity score for a word pair is computed using PMI-weighted Needleman-Wunsch algorithm. We transform the word similarity score using sigmoid function to yield a score between 0 and 1.0.

SCA We experimented with SCA (Sound Class Based Phonetic Alignment) word distance score (List et al., 2016) as an additional feature in our SVM model and found that inclusion of this feature improves the performance of cognate clustering systems. The SCA distance score is computed using the LingPy library (List et al., 2017a).

All the above features are widely used in cognate identification papers cited in sections 1 and 2. All the string similarity features are computed on words represented in ASJP code consisting of symbols on standard QWERTY keyboard. The ASJP code consists of 41 symbols that is used to represent common sounds of the world's languages. As such it collapses some distinctions between similar sounds such as using a single 'r' symbol for all the rhotic sounds. In this paper, we used LingPy library to convert IPA symbols to ASJP symbols. Our SVM model is implemented using scikit-learn (Buitinck et al., 2013). The trained SVM model is then used to predict the confidence scores for all the word pairs having the same meaning.

4 Clustering algorithms

In this section, we motivate and describe the two sd-CRP algorithms followed by InfoMap and UPGMA clustering algorithms.

4.1 Motivation for CRP

In the traditional CRP, the probability that a new customer i sits at a table already filled with customers is proportional to the number of customers sitting at the table. The probability that the new customer sits at a new table is proportional to α. Blei and Frazier (2011) extended the traditional CRP model to a distance-dependent CRP model (dd-CRP) where customer i sits with a different customer j with a probability proportional to $f(d_{ij})$ where f is a decay function and d_{ij} is the distance between customers i and j. The new customer can sit by itself with a probability proportional to α. The dd-CRP formulation forms clusters through connections between the customers. This property to form clusters depending on the data is directly relevant for inferring cognate clusters from a word pair distance matrix.

In a later paper, Socher et al. (2011) introduced a similarity dependent CRP (sd-CRP) algorithm that can handle arbitrary similarities between two customers. Socher et al. (2011) showed that their sd-CRP variant performs better than dd-CRP when clustering MNIST digits dataset and Newsgroup articles. A customer is a word in the context of cognate identification. We describe the two variants of sd-CRP – ns-CRP and sb-CRP – that work directly with a similarity matrix S in the next section.

4.2 sd-CRP algorithms

Given a word similarity matrix $S \in \mathbb{R}^{N \times N}$ and α, the CRP algorithm clusters N elements into K clusters where $1 <= K <= N$.

4.2.1 ns-CRP

The algorithm starts by placing each word into its own cluster. At each step, the algorithm assign a word w_i to the cluster C that has the highest net similarity with w_i which gives the name to the algorithm. We define net similarity as

Algorithm 1 ns-CRP

```
Input: S, $\alpha$
Ouput: Cluster assignments
  1. Initialize each word into its own cluster and set $\alpha$ to 0.1.
  2. Repeat until convergence:
     • For each word $w_i$
        – Remove $w_i$ from its cluster.
        – Compute the net similarity $s_{ik}$ between $w_i$ to all words in a cluster $k$.
        – If $\arg\max_k s_{ik} < \alpha S(w_i, w_i)$ assign $w_i$ to a new cluster.
        – Else, assign $w_i$ to the cluster $k$ where $k = \arg\max_k s_{ik}$.
     • Sample $\alpha$ using a Metropolis-Hastings step
```

$\sum_{j=1}^{|C|} S(w_i, w_j)$. We call the algorithm ns-CRP after the net similarity criterion used to perform cluster assignments. w_i is assigned to a new cluster if $\alpha S(w_i, w_i)$ is greater than any of the similarities with the existing clusters. Any empty clusters remaining at the end of an iteration are removed. The cluster inference procedure is summarized in Algorithm 1.

Algorithm 2 sb-CRP

```
Input: S, $\alpha$
Ouput: Cluster assignments
  1. Initialize each word to its own cluster and set $\alpha$ to 0.1.
  2. Repeat until convergence:
     • For each word $w_i$
        – Remove the outgoing link from $w_i$.
        – Compute the net similarity $s_{ik}$ between $w_i$ and the words in the set returned by SitBehind($w_k$).
        – If $\arg\max_k s_{ik} < \alpha S(w_i, w_i)$ assign $w_i$ to a new cluster.
        – Else, link $w_i$ to a word $w_k$ where $k = \arg\max_k s_{ik}$.
     • Sample $\alpha$ using a Metropolis-Hastings step
```

4.2.2 sb-CRP

The sd-CRP variant of Socher et al. (2011) forms a directed link from word w_i to a different word w_{-i} based on the SITBEHIND function. We call this variant of sd-CRP algorithm as sb-CRP after SitBehind function. The function SitBehind(w_i) is recursive in nature and returns the set of words from which there is a path to w_i including itself. A directed link between w_i to itself indicates that there is no path from w_i to any other word and that w_i is in its own cluster. The probability of forming a directed link from w_i and w_j is proportional to the sum of the similarity between w_i and all the words in the set returned by SitBehind(w_j). The weight for linking w_i to itself is computed as $\alpha S(w_i, w_i)$. The sb-CRP is summarized in Algorithm 2.

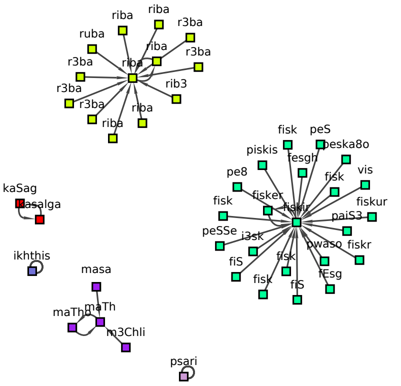


Figure 1: sb-CRP clustering for meaning *fish*. Vertices (words) with the same color are cognates.

We present the result of application of sb-CRP algorithm to meaning *fish* in figure 1. The algorithm places the words correctly in their own clusters. The algorithm forms singleton clusters by forming self-loops. For instance, the algorithm links Ancient Greek `ikhthis` to itself thus, placing the word in its own cluster. When two words belonging to Bihari and Oriya are highly similar `maTh` $\sim$ `maTho` then, the algorithm links both the words to each other forming a cycle.

4.2.3 Underlying objective

Given K clusters out of which n are non-singleton, algorithm 1 maximizes the following objective where k is the cluster index.

$$\sum_{k=1}^{n} \sum_{(i,j)\in k} S(w_i, w_j) - \sum_{k=n+1, i\in k}^{K} \alpha S(w_i, w_i) \quad (1)$$

In the initial step, the objective in equation 1 is $-\alpha \sum_i S(w_i, w_i)$ which increases until there is no change in the cluster reassignments. The objective for algorithm 2 is similar to equation 1 and only differs in the positive part due to SitBehind function. We use the above objective to sample α which is explained below. We observe that the objective function given in equation 1 is similar to the CRP extension to K-Means (DP-Means) proposed by Kulis and Jordan (2011) who show that the DP-means algorithm converges to a local optimum.

4.2.4 Sampling α

We sample α using a Metropolis-Hastings step. We will assume an exponential prior for α with rate parameter 10. We assume an exponential prior since α should be greater than zero and the support for the exponential distribution is $\mathbb{R}^+$. α is sampled through a Metropolis-Hastings step at the end of each iteration. We use an asymmetric multiplier proposal $q(\alpha^*|\alpha) = \alpha \cdot e^{\varepsilon(u-0.5)}$ where $u(\in [0,1])$ is a uniform random number to propose a new α^*. The Hastings ratio for a multiplier proposal is $\varepsilon(u-0.5)$ where ε $(=1)$ is the tuning parameter that controls the range of proposed α^* (Lakner et al., 2008). Since we sample α on fixed cluster assignments, the likelihood ratio is equal to $\frac{\alpha^*}{\alpha}$. The prior ratio is equal to $\frac{\exp(\alpha^*)}{\exp(\alpha)}$.

In this paper, we run both the sd-CRP algorithms by setting the initial value of α to 0.1 and running the algorithms for 100 iterations. We found that the algorithm converges within the first ten iterations (see section 7.4). The algorithms take less than three hours to run for the Austronesian language family. We report the final iteration's B-cubed F-scores and ARI scores (see section 5.2) for each dataset.

4.3 Other Clustering algorithms

UPGMA The variant of St Arnaud et al. (2017) applied a ReLU transformation ($max(0, s)$) to the pairwise similarity matrix S such that the matrix consists only of positive similarity scores. In the initial step, each word is placed in its own cluster. The mutual score between two clusters is computed as the average of the similarity scores between all the word pairs. In each step, the algorithm merges two clusters with the highest pairwise score. The merging process is only stopped when no two clusters have positive average similarity score.

InfoMap is an information-theoretic based clustering algorithm that uses random walks to detect clusters in a network (Rosvall and Bergstrom, 2008). We transform the similarity matrix into a distance matrix by applying a sigmoid transformation then subtracting the matrix values from 1.0. Then, we apply a pre-defined threshold to form a disconnected graph. Finally, we supply the disconnected graph as input to the InfoMap algorithm to infer clusters. We also experimented with the threshold during cross-validation experiments on the training dataset and found that a threshold of 0.57 yielded slightly higher performance than a threshold of 0.5.

5 Materials and Evaluation

In this section, we describe the datasets and cluster evaluation metrics.

5.1 Datasets

Training dataset Wichmann and Holman (2013) and List (2014) compiled cognacy annotated multilingual word lists for subsets of families from various scholarly sources such as comparative handbooks and historical linguistics' articles. The detailed references to all the datasets are given in Jäger et al. (2017). Below, we provide the number of languages/number of meanings in each language group in parantheses.

- Afrasian (21/40), Kadai (12/40), Kamasau (8/36), Lolo-Burmese (15/40), Mayan (30/100), Miao-Yao (6/36), Mixe-Zoque (10/100), Mon-Khmer (16/100), Bai dialects (9/110), Chinese dialects (18/180), Japanese (10/200), ObUgrian (21/110; Hungarian excluded from Ugric sub-family).

We extracted a total of 48,389 cognate pairs (positive) and 51,452 non-cognate pairs (negative) for training our SVM model.

Test datasets We test our clustering algorithms on word lists belonging to four language families given in table 2.

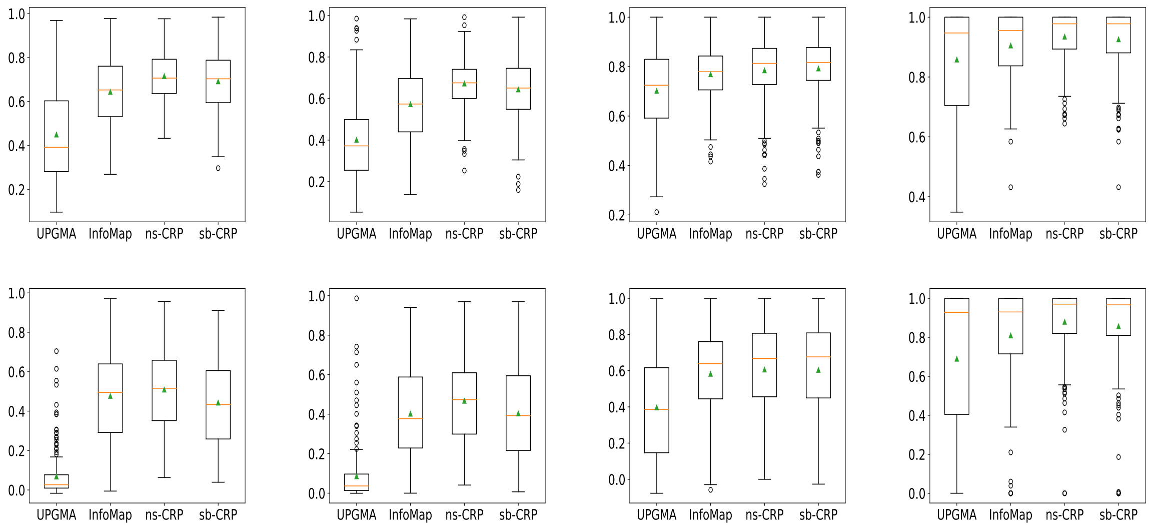
Dataset	Meanings	Languages	Source
Austronesian	210	395	Gray et al. (2009)
Austro-Asiatic	200	122	Sidwell (2015)
Indo-European	208	52	Bouckaert et al. (2012)
Central Asian dialects	183	88	Mennecier et al. (2016)

Table 2: The second, third, and fourth columns show the number of number of meanings, languages, and the source of each dataset respectively.

5.2 Evaluation

We use B-cubed F-score (Amigó et al., 2009) and Adjusted Rand Index (Hubert and Arabie, 1985) to evaluate the quality of the inferred clusters.

B-cubed F-scores are defined for each individual item (word) as follows. The precision for an item is defined as the ratio between the number of cognates in its cluster to the total number of items



(a) Austronesian

(b) Austro-Asiatic

(c) Indo-European

(d) Central Asian

Figure 2: The B-cubed F-scores are shown in the top row. The bottom row shows the ARI scores for each of the datasets. The horizontal bar shows the median score and the mean of the scores is shown by ▲.

in its cluster. The recall for an item is defined as the ratio between the number of cognates in its cluster to the total number of expert labeled cognates. Finally, the B-cubed F-score for a meaning is computed as the harmonic mean of the items' average precision and recall. The B-cubed F-score for the whole dataset is computed as the average of the B-cubed F-scores across all the meanings.

Adjusted Rand Index (ARI) is a chance corrected version of rand index (Hubert and Arabie, 1985). The ARI scores are in the range of -1 to $+1$. A score of 0 indicates that the obtained clusters are randomly labelled whereas a score $+1$ indicates perfect match between the two clusters. The ARI score is zero whenever the gold standard groups all the words belonging to the same meaning slot (e.g. words for meaning *name* are cognate across the daughter Indo-European languages) as one cluster, whereas the B-cubed F-score is not zero in such a case.

6 Results

6.1 F-scores and ARI

We visualize the B-cubed F-scores and ARI scores in figure 2. The spread of the F-scores and ARI scores suggest that InfoMap and sd-CRP variants are better than UPGMA in the case of all the datasets except for the Central Asian dataset. The box plots for InfoMap are similar to the box plots of sd-CRP variants across all the language families. InfoMap and sd-CRP variants have shorter width boxes than those of UPGMA across all the families. All the algorithms show the lowest performance in terms of both F-scores and ARI scores on the Austro-Asiatic dataset. Based on mean F-scores and ARI scores across all the four language families, we determine the ns-CRP algorithm to be the winner.

6.2 Size of inferred clusters

Method	Austro-Asiatic	Austronesian	Central Asian	Indo-European
UPGMA	0.194	0.186	0.722	0.659
InfoMap	0.438	0.617	0.8	0.753
ns-CRP	0.609	0.77	0.833	0.816
sb-CRP	0.564	0.716	0.833	0.817

Table 3: Pearson's R between number of predicted clusters and number of clusters in the gold standard data. The best correlation for each language family is shaded in light gray.

Apart from evaluating the cluster quality using B-cubed F-scores and ARI scores, we compare the number of inferred clusters by each algorithm against the number of clusters given in the gold standard data using Pearson's R. We present the results of Pearson's correlation in table 3. The Pearson's correlation between the number of predicted clusters and the number of gold clusters shows that the sd-CRP variants are successful at retrieving the right number of clusters when compared to UPGMA. InfoMap comes close to both sd-CRP variants' performance only in the case of the Central Asian languages dataset. The ns-CRP algorithm is the winner at being the best predictor of cluster sizes since it predicts clusters of sizes close to those given in the gold standard in the case of Austro-Asiatic and Austronesian datasets and shows same performance as sb-CRP in the case of the Central Asian dialects dataset.

7 Discussion

In this section, we discuss the effect of feature selection and initial value of α on the performance of sd-CRP algorithms. We verify the effect of missing data on all the clustering algorithms and present the results. Finally, we analyze the working of sd-CRP algorithms.

7.1 Feature ablation

To ascertain which word similarity features contribute the most to the performance of the ns-CRP algorithm, we trained three simpler SVM models and evaluated the quality of the inferred clusters using these models. The first model HK uses only orthographic features. The second model uses the PMI word similarity as an additional feature to the HK model. The third model uses SCA word similarity as an additional feature to the HK model. The results presented in previous section showed that ns-CRP performs the worst on Austronesian and Austro-Asiatic datasets.

Therefore, we present the cluster evaluation results only for these two datasets in table 4. The HK model yields high F-scores for both the datasets. Addition of PMI or SCA as an additional feature always improves both F-scores and ARI scores. In fact, including both PMI and SCA as features yields the best results even if the improvement is marginal in the case of the Austro-Asiatic dataset. We note that we observe similar trends for the rest of the datasets. We do not present the results for other datasets due to space constraints. Finally, the ablation experiments suggest that including both data-driven PMI and linguistically guided SCA as features gives the best results at cognate clustering.

System	F-score	ARI
	Austronesian	
HK	0.675 ± 0.111	0.416 ± 0.189
HK+PMI	0.706 ± 0.126	0.489 ± 0.20
HK+SCA	0.683 ± 0.111	0.443 ± 0.193
HK+PMI+SCA	0.715 ± 0.111	0.509 ± 0.193
	Austro-Asiatic	
HK	0.638 ± 0.117	0.389 ± 0.185
HK+PMI	0.651 ± 0.139	0.435 ± 0.219
HK+SCA	0.666 ± 0.117	0.441 ± 0.197
HK+PMI+SCA	0.672 ± 0.127	0.467 ± 0.213

Table 4: Results of feature ablation experiments on Austronesian and Austro-Asiatic datasets.

7.2 Effect of lexical coverage

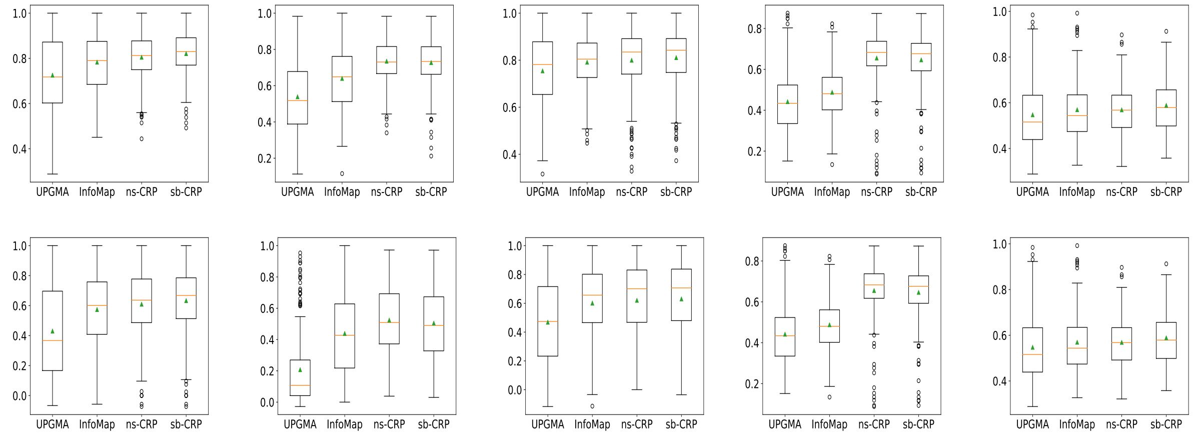
In this subsection, we investigate the effect of missing data on the clustering algorithms. In the case of the Austronesian dataset, less than 50% of the languages have word forms attested in 70% of the meanings. The situation is slightly better in the case of Austro-Asiatic with more than 80% of the languages having meanings attested in 70% of the meanings.

In a separate paper, Rama et al. (2018) presented pruned datasets for five different language families – Pama-Nyungan and Sino-Tibetan in addition to Austronesian, Austro-Asiatic, and Indo-European – consisting of only those languages that show the highest mutual lexical coverage. For each dataset, the authors pruned any language which has less than 75% mutual attestations with the rest of the languages. We attempted to prune the Central Asian dataset but found that we could only exclude a single dialect which has less than 50% attestation. Therefore, we did not include the Central Asian dataset in our experiments. The statistics of the pruned datasets is given in table 5.

Family	Meanings	Languages
Austronesian	210	45
Austro-Asiatic	200	58
Indo-European	208	42
Pama-Nyungan	183	67
Sino-Tibetan	110	64

Table 5: The dataset shows the number of meanings and languages in the pruned datasets.

The results of this experiment are visualized in figure 3. The sd-CRP algorithms perform better than UPGMA and InfoMap in the case of Pama-Nyungan and Austro-Asiatic datasets. There



(a) Austronesian (b) Austro-Asiatic (c) Indo-European (d) Pama-Nyungan (e) Sino-Tibetan

Figure 3: The top row shows the B-cubed F-scores and the bottom row shows the ARI scores for pruned datasets of five language families.

seems to be no difference in the performance of all the algorithms in the case of the Sino-Tibetan dataset. There is no difference between sd-CRP and InfoMap algorithms in the case of the Austronesian dataset. Although the mean B-cubed F-scores indicate that there is no difference between the algorithms in the case of the Indo-European dataset, the spread of the box plots suggests that non-UPGMA algorithms perform better than UPGMA. The B-cubed F-scores are not decisive in the case of the Indo-European dataset, whereas the ARI score clearly shows that non-UPGMA perform better than UPGMA. In conclusion, both the sd-CRP algorithms perform at least as good or better than InfoMap algorithm in the case of pruned datasets.

7.3 Effect of initial alpha

Family	F-score	ARI
Austro-Asiatic	0.735 ± 0.119	0.524 ± 0.217
Austronesian	0.805 ± 0.109	0.609 ± 0.242
Indo-European	0.8 ± 0.135	0.62 ± 0.278
Pama-Nyungan	0.655 ± 0.141	0.354 ± 0.174
Sino-Tibetan	0.569 ± 0.114	0.276 ± 0.17

Table 6: The mean and standard deviation of the F-scores and ARI scores for $\alpha = 0.001$ on pruned datasets.

In this experiment, we test the sensitivity of ns-CRP algorithm to the initial α by initializing α to 0.001, 0.01, and 1.0. We hypothesize that our sampling step makes the algorithm robust to the initial value of α. We run the ns-CRP clustering algorithm for 100 iterations for different starting values of α on each of the pruned datasets. The results of the experiment are given in table 6 for $\alpha = 0.001$. The B-cubed F-scores and ARI scores are quite similar for other initial values of α, and therefore we do not present those results to avoid repetition. These results suggest that the ns-CRP algorithm is not sensitive to the value of initial α.

7.4 Convergence of ns-CRP

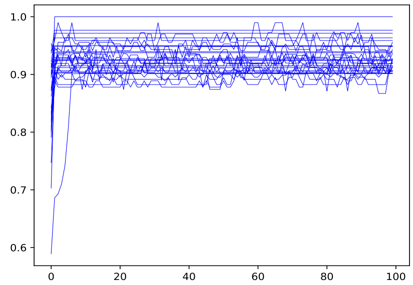


Figure 4: Plot showing the convergence of the sd-CRP algorithm for 30 meanings from the Indo-European dataset.

Here, we investigate the stability of the ns-CRP algorithm by plotting the B-cubed F-scores against the number of iterations for 30 random meanings from the Indo-European dataset in figure 4. The plot shows that the ns-CRP algorithm quickly moves from an initial configuration with low F-score to a configuration that has high F-scores within the first 20 iterations. We observe similar behaviour of ns-CRP in the case of other language families. In conclusion, the plot shows that the quality of the clusters inferred by the ns-CRP algorithm achieves a high F-score. Moreover, the cluster quality does not change drastically after reaching a local optimum.

7.5 Analysis of sd-CRP algorithms

In this subsection, we analyze the difference in the behaviours of sd-CRP algorithms. If w_i and w_j are cognate and w_j and w_k are cognate, then all the three words are cognate with each other which follows from the definition of cognacy. The sb-CRP algorithm captures this cognacy relation through the SitBehind function. During cluster formation, w_i only has to connect to a word that might have no other words other than itself sitting behind it. We hypothesize that the sb-CRP algorithm would be more efficient at identifying partial cognates where only part of the lexical material is cognate with another word. An example of a partial cognate is the meaning of *meat* in *sweetmeat* which is cognate with Swedish *mat* 'food' (Campbell, 2004). In contrast, the ns-CRP algorithm is stricter than sb-CRP algorithm in that a word is assigned to the cluster with which it has the highest net similarity. If a word has net similarity of zero with all the existing clusters, then, the word would form its own cluster since $\alpha S(w_i, w_i)$ is always positive.

8 Conclusion

We presented and compared the performance of two similarity dependent Chinese Restaurant process algorithms at the task of automated cognate detection for six different language families. The sensitivity experiments suggested that the sd-CRP algorithms is not sensitive to initial α and missing data. The feature ablation experiments suggest that the inclusion of PMI and SCA features improve the performance of the sd-CRP algorithms. We conclude that the sd-CRP algorithms perform better than the existing clustering algorithms across multiple settings.

As future work, we plan to include language relatedness as features into SVM training and also train the SVM classifier in an unsupervised fashion using the sd-CRP algorithms.

Acknowledgments

The author thanks the anonymous reviewers for the comments which helped improved the paper. The author is supported by BIGMED project (a Norwegian Research Council LightHouse grant, see bigmed.no). The algorithms were designed when the author took part in the ERC Advanced Grant 324246 EVOLAEMP project led by Gerhard Jäger. All these sources of support are gratefully acknowledged.

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.

David M Blei and Peter I Frazier. 2011. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488.

Phil Blunsom and Trevor Cohn. 2011. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 865–874. Association for Computational Linguistics.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 110(11):4224–4229.

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

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Lyle Campbell. 2004. *Historical Linguistics: An Introduction*. Edinburgh University Press, Edinburgh.

Will Chang, Chundra Cathcart, David Hall, and Andrew Garrett. 2015. Ancestry-constrained phylogenetic analysis supports the Indo-European steppe hypothesis. *Language*, 91(1):194–244.

Michael A. Covington. 1996. An algorithm to align words for historical comparison. *Computational Linguistics*, 22(4):481–496.

Samuel J Gershman and David M Blei. 2012. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 673–680. Association for Computational Linguistics.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2011. Producing power-law distributions and damping word frequencies with two-stage language models. *Journal of Machine Learning Research*, 12(Jul):2335–2382.

Russell D Gray, Alexei J Drummond, and Simon J Greenhill. 2009. Language phylogenies reveal expansion pulses and pauses in pacific settlement. *science*, 323(5913):479–483.

David Hall and Dan Klein. 2011. Large-scale cognate recovery. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 344–354. Association for Computational Linguistics.

Bradley Hauer and Grzegorz Kondrak. 2011. Clustering semantically equivalent words into cognate sets in multilingual lists. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 865–873, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

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

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

Gerhard Jäger and Johann-Mattis List. 2017. Using ancestral state reconstruction methods for onomasiological reconstruction in multilingual word lists. *Forthcoming, Language Dynamics and Change*.

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. Association for Computational Linguistics.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *North American Chapter Of The Association For Computational Linguistics*, pages 1–8. Association for Computational Linguistics Morristown, NJ, USA.

Grzegorz Kondrak. 2002. *Algorithms for language reconstruction*. Ph.D. thesis, University of Toronto, Ontario, Canada.

Grzegorz Kondrak. 2005. Cognates and word alignment in bitexts. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 305–312.

Grzegorz Kondrak. 2009. Identification of cognates and recurrent sound correspondences in word lists. *Traitement Automatique des Langues et Langues Anciennes*, 50(2):201–235.

Brian Kulis and Michael I Jordan. 2011. Revisiting k-means: New algorithms via Bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*.

Clemens Lakner, Paul Van Der Mark, John P Huelsenbeck, Bret Larget, and Fredrik Ronquist. 2008. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Systematic biology*, 57(1):86–103.

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

Johann-Mattis List. 2012b. SCA: phonetic alignment based on sound classes. In *New Directions in Logic, Language and Computation*, pages 32–51. Springer.

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

Johann-Mattis List, Simon Greenhill, and Robert Forkel. 2017a. Lingpy. a python library for quantitative tasks in historical linguistics.

Johann-Mattis List, Simon J. Greenhill, and Russell D. Gray. 2017b. The potential of automatic word comparison for historical linguistics. *PLOS ONE*, 12(1):1–18.

Johann-Mattis List, Philippe Lopez, and Eric Bapteste. 2016. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Berlin, Germany. Association for Computational Linguistics.

Philippe Mennecier, John Nerbonne, Evelyne Heyer, and Franz Manni. 2016. A central asian language survey. *Language Dynamics and Change*, 6(1):57–98.

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.

Taraka Rama. 2015. Automatic cognate identification with gap-weighted string subsequences. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*, pages 1227–1231.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1018–1027.

Taraka Rama, Prasant Kolachina, and Sudheer Kolachina. 2013. Two methods for automatic identification of cognates. *QITL*, 5:76.

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)*, volume 2, pages 393–400.

Taraka Rama, Johannes Wahle, Pavel Sofroniev, and Gerhard Jäger. 2017. Fast and unsupervised methods for multilingual cognate clustering. *arXiv preprint arXiv:1702.04938*.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 12–21. Association for Computational Linguistics.

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

Paul Sidwell. 2015. Austroasiatic lexical data set for phylogenetic analyses 2015 version.

Kairit Sirts, Jacob Eisenstein, Micha Elsner, and Sharon Goldwater. 2014. Pos induction with distributional and morphological information using a distance-dependent chinese restaurant process. In *ACL (2)*, pages 265–271.

Benjamin Snyder, Regina Barzilay, and Kevin Knight. 2010. A statistical model for lost language decipherment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1048–1057. Association for Computational Linguistics.

Richard Socher, Andrew L Maas, and Christopher D Manning. 2011. Spectral chinese restaurant processes: Nonparametric clustering based on similarities. In *AISTATS*, pages 698–706.

Robert R Sokal and Charles D Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438.

Adam St Arnaud, David Beck, and Grzegorz Kondrak. 2017. Identifying cognate sets across dictionaries of related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2518.

Morris Swadesh. 1952. Lexico-statistic dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos. *Proceedings of the American philosophical society*, 96(4):452–463.

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

A Supplemental Material

The code and data used in this paper are uploaded as a zip file along with this paper. In addition, they are available for download at: `https://github.com/PhyloStar/sd-CRP-cognates`