IETE Journal of Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tijr20

A Language Lending Itself: Mapping Clusters of Contextually Close Cognates in Indo-European Languages

Sarthak Rastogi & Manpreet Kaur

To cite this article: Sarthak Rastogi & Manpreet Kaur (2023): A Language Lending Itself: Mapping Clusters of Contextually Close Cognates in Indo-European Languages, IETE Journal of Research, DOI: 10.1080/03772063.2023.2245370

To link to this article: https://doi.org/10.1080/03772063.2023.2245370

Published online: 30 Aug 2023.

Submit your article to this journal

Article views: 33

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tijr20

IETE JOURNAL OF RESEARCH
https://doi.org/10.1080/03772063.2023.2245370

Check for updates

A Language Lending Itself: Mapping Clusters of Contextually Close Cognates in Indo-European Languages

Sarthak Rastogi and Manpreet Kaur

Department of Computer Science and Technology, Manav Rachna University, Haryana, India

ABSTRACT
A key method for understanding the evolution of languages is to look for words which share etymological roots across languages. These words, called lexical cognates, allow historical linguists to group languages into families and study their structural history. Existing research on cognate clustering is chiefly based on analyses performed on lexical databases which limits the scope for exploring new cognates. Additionally, while the Indo-European family is a transcontinental one, its cognate analyses are largely limited to a number of European languages. In this research work we build a new dataset using word embeddings to search for cognates using phonetic matching between translations of words in different languages across the Indo-European family. The context-dependent positioning of words in word embeddings allows for comparisons with contextually similar words and hence clustering using unsupervised learning algorithms. We successfully find significant distinctions between these "contextually close clusters of cognates" among languages across the Indo-Iranian, Romance, and Germanic families. We release this novel method to demonstrate which words a language is more likely to lend to another.

KEYWORDS
Word embeddings; Cognates; Unsupervised clustering; Linguistic analysis; NLP

1. INTRODUCTION

The Indo-European language family descended from what is reconstructed today as the Proto-Indo-European language (PIE), which was spoken circa 4,500 - 2,500 BCE. During the Indo-European migrations that occurred by 3,000 - 2,000 BCE, the speakers of PIE separated, and over the millennia the various dialects they spoke transformed into the modern Indo-European languages. These languages comprise, but are not limited to, Hindi-Urdu, English, French, German, Latin, Marathi, Punjabi, and Spanish; together, they are spoken today by 40% of the world's population. A subset of the family tree for PIE is shown in Figure 1.

To understand the evolution of these languages over time, and to group them into families, historical linguists compare them and identify words which may have been passed down to them from the same language, or which they may have lent to each other. These words, called lexical cognates, are key to the study of the structural history of languages. Traditionally, the comparative method has been used for identifying cognates; however, the method can take decades to complete this task for sizeable language families when performed by hand [1].

This has motivated much research over the last two decades to automate this process; scholars have come up with both programmatic and machine learning methods to identify clusters of cognates. In this paper we introduce a new methodology to efficiently create cognate datasets for any two given languages using word embeddings and hence cluster these "contextually close cognates" using unsupervised learning algorithms. We term our method "L2C4" as an abbreviation for the title of this paper.

Existing literature on the topic of clustering cognates together is chiefly based on analyses performed using lexicostatistical databases. These are wordlists that often lack information on the relationships between words. Much of the work is limited to clustering the cognates already derived in the databases. These papers also limit themselves to a small set of European languages, while the Indo-European family transcends countries and even continents; this leaves much scope for making comparisons across the three largest language groups in the family: Romance, Germanic and Indo-Iranian.

The motivation behind introducing the idea of clustering contextually close cognates together is to understand the following: throughout periods of exchanges between speakers of different languages in history, when a language lent itself to another, what type of words were more frequently passed on. Our method uses word embeddings as a dataset because they contain words positioned

© 2023 IETE

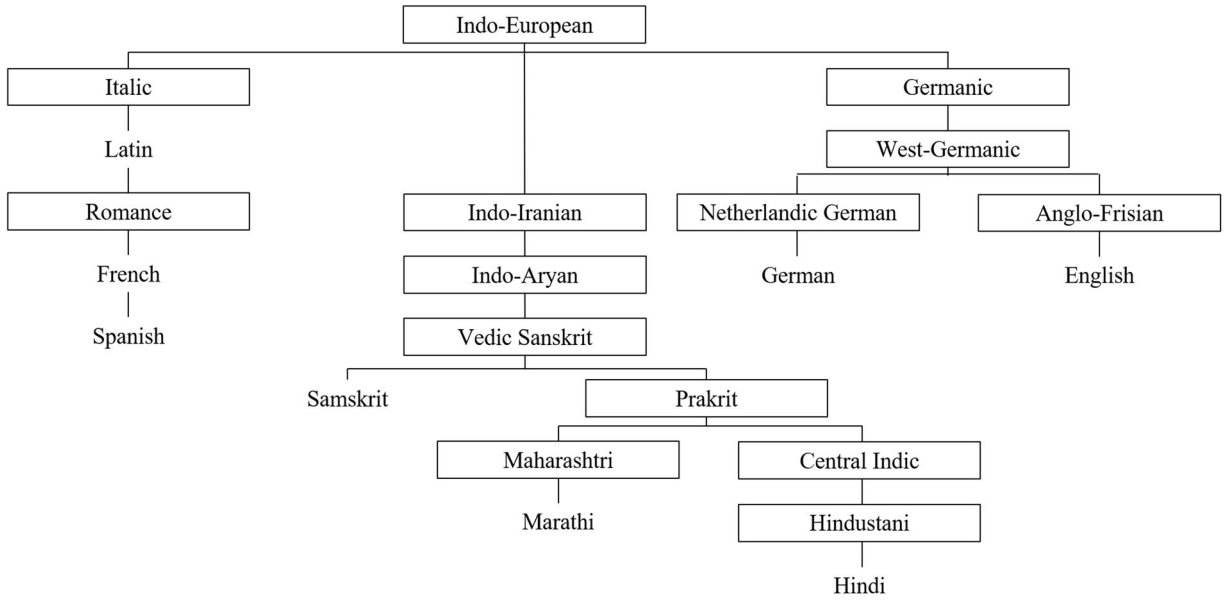


Figure 1: A subset of the Proto Indo-European language tree, showing the origins of languages we include in this study [2].

closer to other words which are used in a context similar to it. We hypothesise that if we find well-defined clusters among our cognate embeddings, they may be groups of words passed on more frequently from one language to another. And if there exist recurring patterns of cognates occurring between multiple pairs of languages, these words might have sociocultural reasons for having survived rifts in language families or having been lent from language to language; this can in turn reveal information on the civilisations who spoke these languages.

The following are the novel contributions of this paper: 1. Introducing a method to create a contextually structured cognate dataset using word embeddings; a dataset that can be readily created makes it easier for further research work to exploit the contextual similarity between words. 2. Comparing Indo-Iranian and European languages among and across each other for cognates in order to expand 3. Producing L2C4 clusters, i.e. clusters of words most likely to be passed down from a language.

The rest of the paper is organised as follows: Section 2 discusses previous work on identifying and clustering cognates. Section 3 introduces the dataset of word embeddings we use and the method used in learning them. Section 4 describes the method followed in preprocessing the raw word embeddings, and their subsequent clustering. Section 5 presents an analysis of the L2C4 clusters obtained, and section 6 further discusses our findings. Section 7 concludes our work.

2. RELATED WORK

2.1 Conventional Comparative Methods in Historical Linguistics

The most common methods for identifying cognates are comparative methods such as those used in [3,4]. The comparative method is used to construct an earlier language by comparing related words and expressions in its descendent languages. The method was originally developed for reconstructing PIE before finding use in the reconstruction of other ancient languages.

2.2 Cognate Identification in Lexical Databases and Wordlists

A majority of the research work on cognate identification is based on lexical databases. Kondrak [5] exploits the noun hierarchy in the WordNet lexical database to measure semantic similarity and the ALINE algorithm to measure phonetic similarity between words from two vocabulary lists. It demonstrates the limitations of measuring word similarity using orthographical approaches [,and makes a credible argument in favour of the effectiveness of phonetic approaches. It also successfully obtains a significantly larger number of cognates owing to the semantic information obtained by the glosses of lexemes in WordNet. Wu and Yarowsky [6] uses a lexical database to compare words for cognacy only if they have the same English translation, thus avoiding the need to compute semantic similarity scores. List [7] uses sequences of words from a multilingual wordlist and computes an encoding for the way they are pronounced.

For each language pair they devise a different scoring scheme and compute the score for each word pair. If the score exceeds a certain threshold for a word pair, they are clustered together as cognates.

2.3 Machine Learning Methods for Cognate Identification and Clustering

Both supervised and unsupervised methods have been employed for cognate clustering. The method proposed in [8] manually compiles a dataset of cognates and false friends in the English and French languages, and uses orthographical similarity measures and machine learning techniques to classify words as cognates and false friends.

Similarly, the method proposed in [9] uses a database of words in Indo-European languages which it classifies into cognates and non-cognates using SVM classifiers on the basis of word similarity features such as minimum edit distance, length of the longest common prefix, etc. Figure 2 contrasts the approach to clustering cognates followed in [9] with our approach.

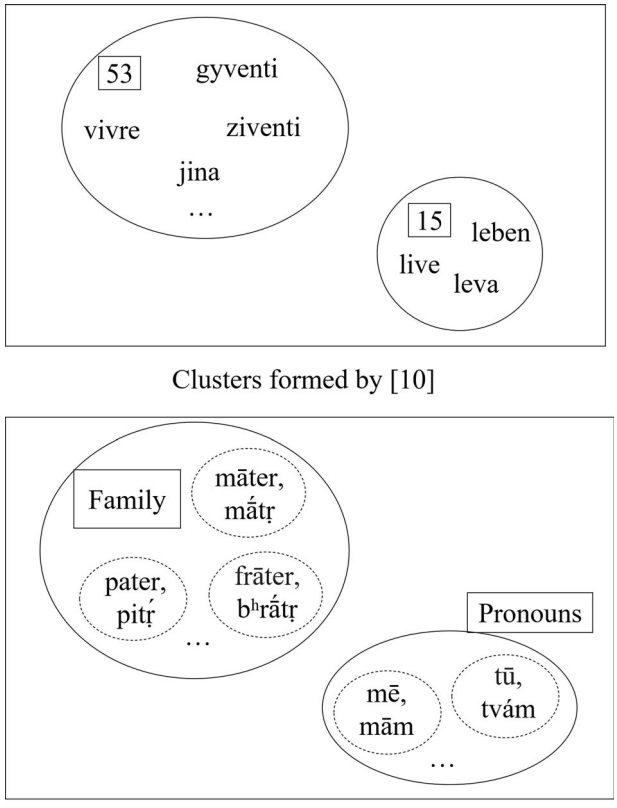


Figure 2: The method used in [9] clusters subsets of phonetically similar cognates together from a set of cognates all with the same meaning but in different languages, while our approach aims to cluster together cognates which are contextually close.

Rama et al. [10] uses lexical databases on 16 different language families to cluster cognates. They compare the performance of unsupervised systems including Pointwise Mutual Information (PMI) and Pair Hidden Markov Model (PHMM), along with LexStat and ALINE. They use B-cubed F-score as the evaluation metric, and find that online trained PMI systems yield the best performance. In a similar survey of phonetic matching algorithms for cognate identification, [11] finds DBN and PHMM to yield optimal 11-point interpolated average precision scores.

The lack of context-awareness in these databases and the scope for improvement in the algorithms applied inspire us to create an end-to-end method for clustering these cognates by the context they occur in.

2.4 Cognate Clustering in Asian Languages

Wu and List [12] focuses on the challenges posed by partial cognates in the context of compounding and derivation specifically in South East Asian languages. It explores current practices of preparing cognate-coded word lists and proposes new approaches to enhance the transparency of cognate annotation. The findings emphasise the significance of careful cognate judgments in 19 Chinese dialects having frequent compounding and derivation, as the choice of conversion method for partial cognate judgments can influence the inferred tree topologies in phylogenetic reconstruction.

Kanojia et al. [13] introduces a novel approach to cognate detection by incorporating cognitive features derived from human readers' gaze behaviour, aiming to enhance the performance of downstream NLP tasks. The authors collect and analyse gaze behaviour data for a small sample of cognates, demonstrating the utility of cognitive features in improving cognate detection. They also leverage predicted cognitive features based on the collected data for a larger sample, achieving significant performance improvements of 10% using the collected gaze features and 12% using the predicted gaze features compared to previous approaches. The authors provide the collected gaze behaviour data, code, and cross-lingual models as open resources.

3. DATASET

3.1 Raw Data Input

We begin with word embeddings pre-trained on the Wikipedia and Gigaword 5 dataset∗. Word embeddings

are high-dimensional vectors of real-valued numbers representing the meaning and context of every word in a vocabulary. The motivation behind using word embeddings as a dataset of words comes from the following structural advantage: they contain words used in a similar context positioned closer to each other. This is not to say that those words are similar in meaning, but that they are used in a similar context in the text dataset the embeddings have been learned from. This property enables us to cluster cognates which are contextually similar.

Another approach unique to – and a benefit of – the dataset we ultimately create is that it equips us with the ability to examine a given number of neighbouring words of each word for phonetic similarity, and hence look for cognacy in the contextual neighbourhood. This is essential to our use case because while cognates may not have the same meaning across languages [8], they are often found to mean words one would use in a similar context. Matching only contextually same words to each other also helps reduce the number of false friends, i.e. words across languages that sound the same but have quite different meanings.

Our word embeddings contain 400,000 words each with a corresponding 50-dimensional vector. They have been learned using GloVe [14], a log-bilinear regression model that uses global matrix factorisation and local context window methods. Another common model is Word2Vec [15], which can either train words on their neighbouring words, or neighbouring words on the original word. However, the semantics learnt using Word2Vec for a certain word rely only on the knowledge of the words surrounding it, allowing Word2Vec to efficiently capture local contextual information about the language but not global information. We hence prefer GloVe over Word2Vec for our use case on account of its ability to capture global word-word occurrence statistics. GloVe has also been shown to perform better on word similarity tasks [14].

*The Stanford GloVe pre-trained embeddings used can be found here: https://nlp.stanford.edu/projects/glove/

The code for our work can be found here: https://github.com/sarthakrastogi/L2C4

4. METHOD

Figure 3 shows the five steps of the proposed methodology in detail.

4.1 Data Pre-processing

We start by lemmatising the English language words. To lemmatise a word is to reduce it to its base form; despite being slower than stemming, which cuts words without analysing the context they're present in, lemmatisation is preferred because it returns an actual language word, a lemma, while stemming returns only word stems which may not translate correctly. We drop recurring lemmas to keep only the first occurrence, which is usually the root word.

Next, we apply named entity recognition to remove proper nouns such as names of places, and then compare our dataset against a database of common names to filter those out as well. This reduces our dataset size to just over 200,000 words.

4.2 Translation and Transliteration

We then translate the embeddings from the English language to Hindi, French, German, Latin, Samskrit and Tamil. In the section that follows, we will be phonetically indexing the words, and the process requires that the words be present in the Roman script. Hence, we Romanise words in the Indian languages from their native scripts using the transliteration methods made available in the IndicNLP library [16]. We also remove definitive (le, la, l' and les) and partitive (du, de la, des, de l', de, d') articles from French words wherever they occur to only retain the noun.

4.3 Pairwise Phonetic Matching

For each word (denoted n_i) in the embeddings of the first language, there exists a corresponding translation (denoted n'_i) in the embeddings of the second language, hence representing an equivalent vector space where nodes n'_i are projections of n_i on the basis of their semantic relationship, as shown in Figure 4. Our aim is to match words from their corresponding spaces N'_i and N'_i to each other if they are phonetically similar, onto C_i, which is the vector space of cognates.

There exist many phonetic indexing algorithms such as Soundex [17], Metaphone [18], etc. which use a set of rules to return a phonetic encoding for a given word. This encoding is a representation of the way the word is pronounced. As shown in [5], these algorithms outperform orthographical algorithms in cognacy identification tasks. In this research work we use the more advanced phonetic matching algorithm Double Metaphone [19]

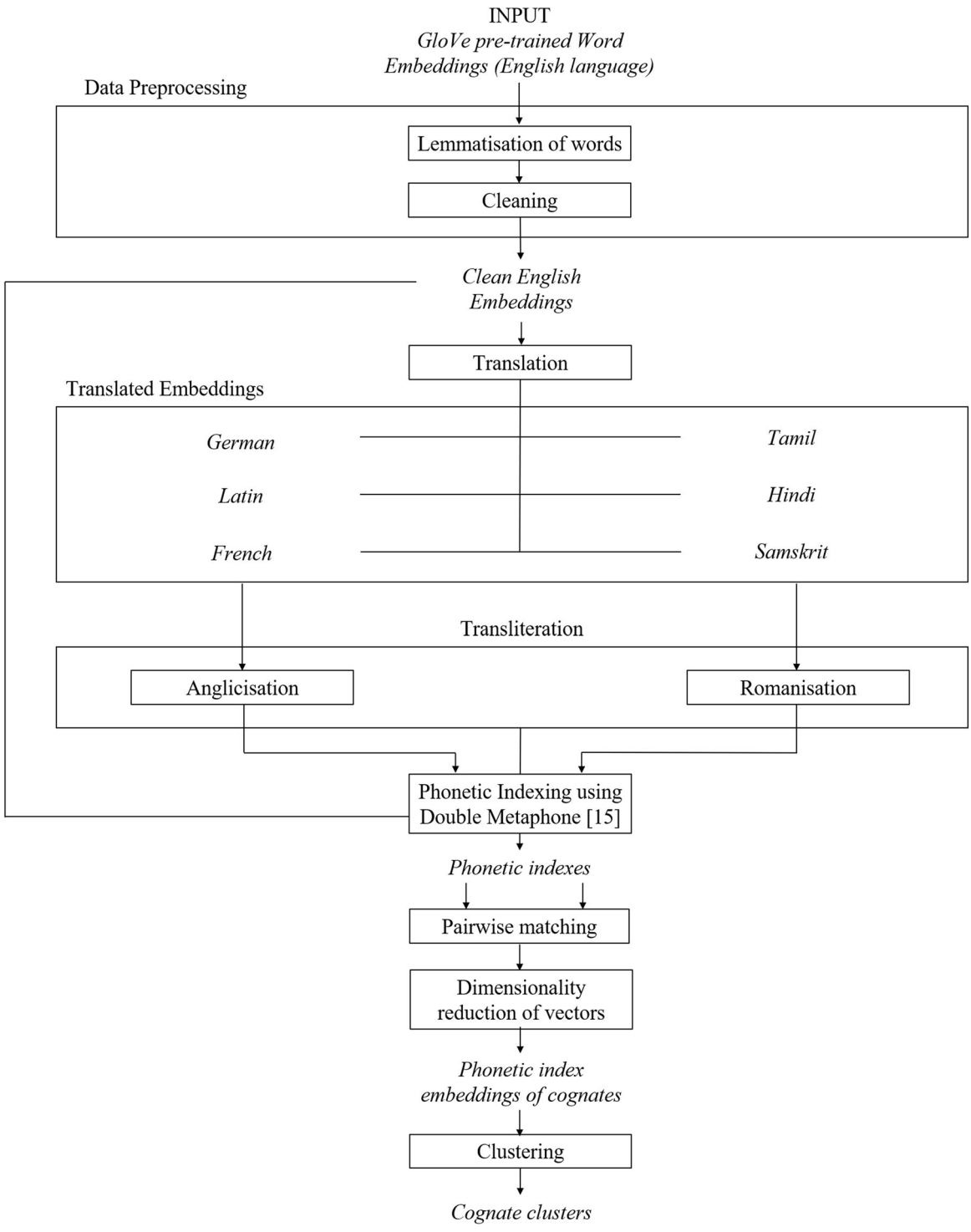


Figure 3: The end-to-end methodology followed in our research work

which introduces a number of performative improvements to Metaphone. For a given word, it returns two codes of four characters each, taking into account that there can be two possible pronunciations.

For each pair of word vectors n and n' in N_i and N'_i respectively, we compute and match their Double Metaphone codes and label the pair as phonetically similar if either of the codes of n matches either of the codes of n'.

The nature of cognates is such that, except in cases of very remotely related languages, they are most likely to either be semantically similar [5], or be used in similar

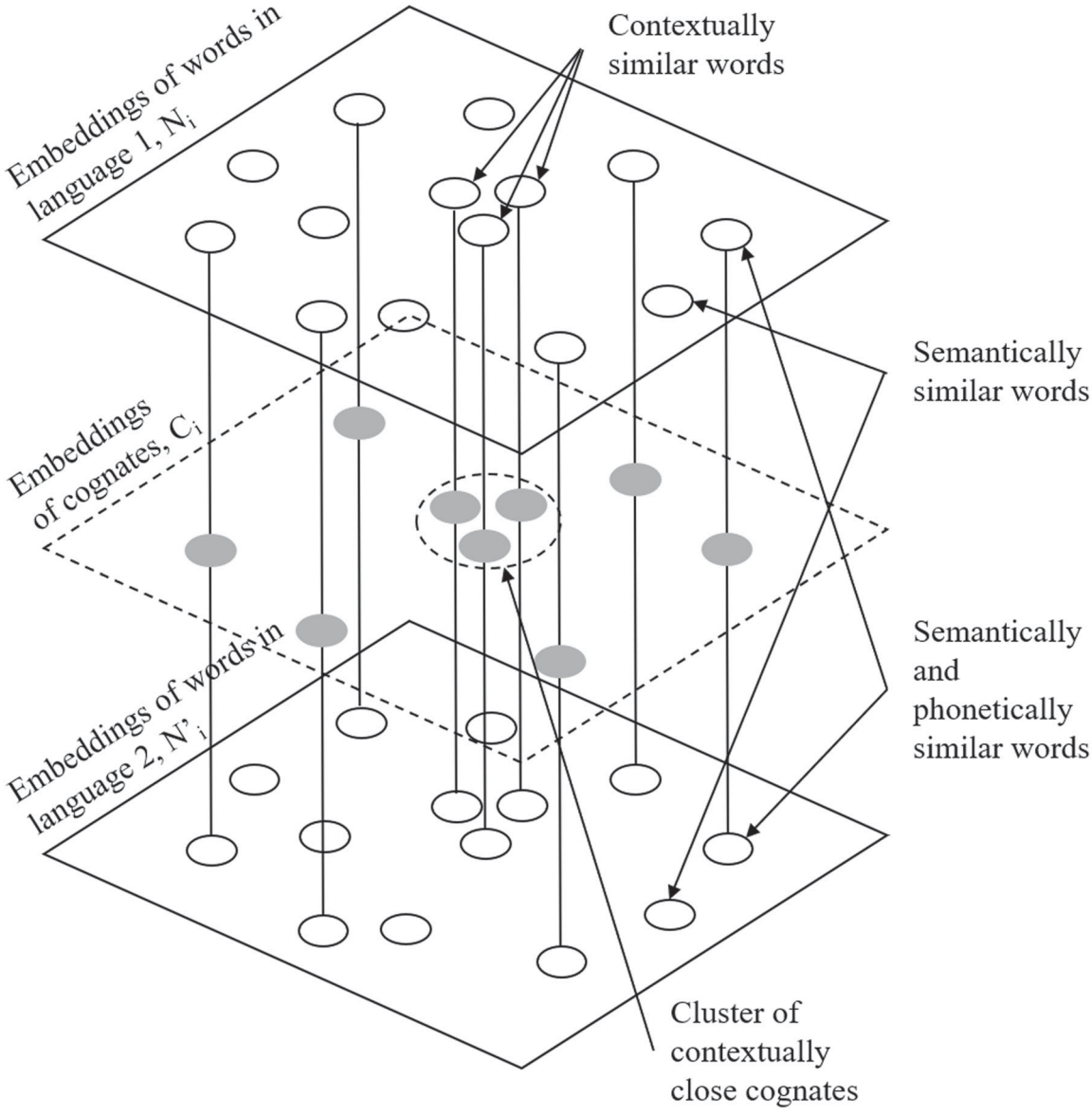


Figure 4: Pairwise phonetic matching in L2C4

context. To capture the latter possibility of cognacy from our embeddings, we also visit the five closest words to n' to check for similarity to any n, and likewise n' to n.

We perform the above comparisons between the following combinations of languages: 1. Samskrit and Latin, 2. Samskrit and Hindi, 3. Hindi and English, 4. German and English, 5. Latin and English, 6. Latin and French, and 7. Hindi and Tamil.

4.4 Dimensionality Reduction

Corresponding to each cognate phonetic index in the obtained embeddings is a 50-dimensional vector. In order to effectively cluster a set of data points, most algorithms require them to be reduced to lower dimensions. We apply Principal Component Analysis (PCA) [20] to our vectors to reduce their dimensionality while preserving a good portion of the variance, followed by t-Distributed Stochastic Neighbour Embedding (t-SNE) [21] to visualise the resulting clusters in two dimensions.

PCA uses orthogonal transformation to convert high dimensional data into the eigenvectors of its covariance matrix, uncorrelated features called principal components (PCs). The first few PCs contain much of the data's variation, and are used to give a lower-dimensional representation for them. To maximise the variation captured by PCs, PCA tries to preserve long pairwise distances; on the other hand, t-SNE aims to preserve short pairwise distances and gives better visualisations for non-linear manifold structured datasets [21]. t-SNE first calculates

the distance between every pair of samples in the dataset, and converts these distances into the normal probability distribution. It then repeats this calculation on random lower dimensional arrangements of the data points, but the distances are converted into the t-distribution. Finally, it minimises the difference between the two probability distributions by running gradient descent using Kullback-Liebler divergence as a cost function.

4.5 Clustering

We next use unsupervised clustering techniques to identify groups in C_i consisting of cognates which are contextually similar to each other. We use only hierarchical and centroid-based clustering techniques in our experiments; word embeddings contain highly sparse vectors, and thus we exclude density-based algorithms such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and OPTICS (Ordering Points To Identify Cluster Structure) from experiments as they fail to give well-formed clusters on sparse data. Below, we describe the clustering algorithms we experiment with.

Affinity Propagation [22] identifies a subset of data points called exemplars, which are the most representative of all the data points, by carrying out message passing between data points each representing the suitability of one point being exemplar to the other, until convergence. Each exemplar, along with the points that it is exemplar to, is grouped into a cluster. To keep the solution at each update step from oscillating, the messages are damped by a factor lying in the range [0.5, 1.0]. Agglomerative Clustering is an hierarchical clustering algorithm that begins with a singleton cluster for each data point, and then continues to merge pairs of clusters which are the closest to each other until either all data points have been clustered together or it has been stopped at a specified number of clusters. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [23] works in two steps: it first condenses the dataset into Clustering Feature (CF) entries; on this structure, called a CF tree, a different clustering algorithm aggregates the leaf nodes into clusters. The K-means algorithm [24] randomly selects a number of centroids among the data points and then optimises their positions until either their values no longer change, or the specified number of iterations has been reached.

5. RESULTS

5.1 Evaluation Metric

We evaluate the quality of the obtained clusters by measuring the silhouette scores [25] of our clusters. If for any sample, d_{intra} denotes the mean distance to all other samples in the cluster, and d_{inter} denotes the mean distance to all other samples in the nearest cluster, we calculate its silhouette coefficient as the following:

$$silhouettecoefficient = \frac{(d_{inter} - d_{intra})}{max(d_{inter}, d_{intra})} \quad (1)$$

The silhouette score is then calculated as the average of the silhouette coefficients of all the samples. It takes values ranging from −1 to 1; a score close to 1 indicates that the obtained clusters are well distinguished, while a score near 0 indicates an insignificant differentiation between clusters and a score nearing −1 indicates that samples have been assigned to wrong clusters.

Table 1: Silhouette scores and number of clusters obtained on clustering cognates between Samskrit and Latin

L2C4 Variant	Silhouette score	Number of clusters
$L2C4_{AP}$	0.5191	11
$L2C4_{AC}$	0.5096	9
$L2C4_{BR}$	0.5096	9
$L2C4_{KM}$	**0.5446**	16

5.2 Cognate Clusters in Similarity Embedding

We produce four variants of the L2C4 method by applying four different clustering algorithms; the variants, termed $L2C4_{AP}$, $L2C4_{AC}$, $L2C4_{BR}$, and $L2C4_{KM}$, employ Affinity Propagation, Agglomerative Clustering, BIRCH, and K-means clustering respectively.

For dimensionality reduction, we apply PCA using 25 principal components, followed by t-SNE, and examine the test scores obtained after clustering on the resulting embeddings. For hyperparameter tuning we experiment with different values of damping factor (for Affinity Propagation) and number of clusters (for Agglomerative Clustering, BIRCH, and K-means), and pick the ones resulting in the highest silhouette scores.

In this paper we are especially interested in studying similarities between transcontinental language pairs. Hence, we start with the Samskrit and Latin cognate embeddings. On applying PCA on their embeddings, we found that 85.44% of the variance was preserved. Table 1 shows the optimal number of cognate clusters and the resulting silhouette scores obtained. We also varied the damping in case of Affinity Propagation, and found that a value of 0.8 was found to be optimal.

In higher dimensional representations of embeddings, words are close to each other in many more directions, and reducing dimensionality often cannot capture them

all. As a result, we also witness much noise consisting of semantically unrelated words in and around most clusters. The visualisations of L2C4 clusters between Samskrit and Latin in Figure 5 display reasonably defined structures.

To judge the performance of the clustering methods, we compare the obtained silhouette scores against a baseline mentioned in [24] which estimates a silhouette score between 0.51–0.70 to indicate a reasonable structure to have been found. In comparison, a score ranging from 0.71–1.0 indicates a strong structure, while a score ranging from 0.26–0.50 indicates a weak structure which can also be artificial, and a score below 0.25 indicates that no significant structure has been found. As shown in Table 2, K-means returns scores over 0.50 in all the language pairs tested above, leading us to the confident conclusion that we have identified meaningful clusters of cognates.

Table 2: Best silhouette scores obtained on clustering cognates between other language pairs

	Samskrit	Hindi	German	Latin	Latin
Language 1					
Language 2					
L2C4 variant	Hindi	English	English	English	French
$L2C4_{AP}$	0.5098	0.4682	0.5209	0.4877	0.5019
$L2C4_{AC}$	0.4363	0.4636	0.4522	0.4291	0.4483
$L2C4_{BR}$	0.4477	0.4831	0.4651	0.4316	0.4256
$L2C4_{KM}$	**0.5258**	**0.5589**	**0.5533**	**0.5570**	**0.5458**

6. DISCUSSIONS

$L2C4_{KM}$ outperforms other variants for all language pairs; k-means displays a consistent ability to cluster cognates on the basis of their respective distance from the cluster centroid. This may be because dimensionality reduction brings points close in the high dimensional space much closer in the lower dimensional representation, and k-means then leverages the Euclidean distance between samples for optimal clustering results.

Upon inspecting individual clusters through the noise, they can be verified to contain contextually similar words. On comparing with the Indo-European vocabulary, we

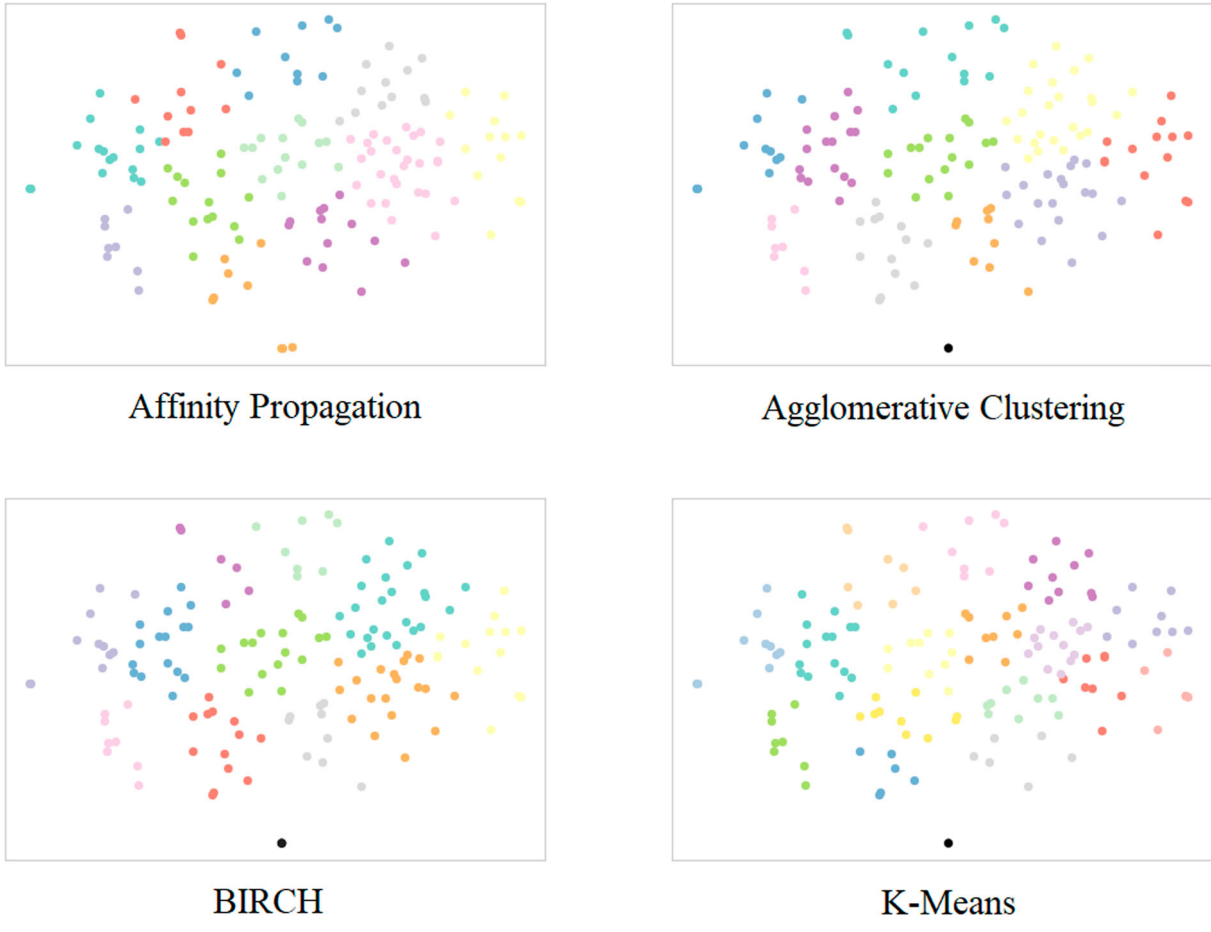


Figure 5: Clusters in the embeddings of cognate indexes between Samskrit and Latin formed using Affinity Propagation, Agglomerative Clustering, BIRCH, and K-means respectively.

confirm our clusters to contain groups of words such as terms for family members, pronouns, numbers, animals and parts of the body. They comprise mostly of nouns, or specifically everyday concepts which can easily be communicated using gestures; the lending of these words hence aligns with the knowledge that cognates were often formed as a result of contact between different language groups since these people would largely communicate with different groups using a combination of speech and gesture to get the meaning of a word across.

Contrary to our assumption behind using Tamil as a control language because of its differentiation from the Indo-European languages we experimented on, comparison between Tamil and Hindi does yield a small number of sparsely scattered cognates. Closer inspection shows that these are indeed more modern words. Owing to the strong evidence present that Tamil belongs to the Dravidian language family, we explain the cognates it has with Hindi to be a result of the contact between the speakers of Dravidian languages in South India and the Indo-Aryan languages in Central and North India.

We expect that linguistic expertise and domain knowledge can contribute to the refinement of the clustering process and the evaluation of the results. Building a large-scale cognate database encompassing multiple language families and historical periods would also greatly benefit the field of historical linguistics and language evolution studies. Such a database could be constructed by aggregating existing lexical resources, collaborating with linguistic experts, and incorporating automated methods like L2C4. The database would facilitate further research, comparative analyses, and the development of advanced algorithms for cognate detection.

This paper was written on the premise that language contact plays a crucial role in the formation of cognates. Investigating the influence of language contact phenomena, such as borrowing, loanwords, and language convergence, on the cognate patterns would provide valuable insights into the dynamics of language evolution. By considering language contact as a factor in the cognate detection process, future research can deepen our understanding of the mechanisms behind cognate formation and identify patterns of lexical borrowing.

7. CONCLUSION

In this paper, we have introduced the L2C4 method and used it to obtain clusters of contextually close cognates in between languages across the Indo-European family. We have identified clusters in language pairs using Double Metaphone, and found that unsupervised clustering using K-means obtained optimal silhouette scores across all language pairs. We have verified the applicability and correctness of the method by comparing our clusters with word groups in the Indo-European Vocabulary, and checking their silhouette scores. We have recognised words used in commonplace conversation to be frequently passed down from language to language. Likewise, the L2C4 method can be used to swiftly compare any two languages for clusters of contextually close cognates.

While our method has presented promising results in detecting cognates, there are several avenues for future exploration and additional work required to enhance the scope and applicability of this research.

Further work using this method may involve datasets of contextually close words in entire language families to track the passage of these words from parent language to child. Due to the automaticity of our method, it can also not address the issue of grouping together false cognates, which are words which sound the same and have similar meanings but have different roots. Methods such as those explained in [8] can possibly be used to address this issue. Techniques such as semantic analysis, word embeddings, or deep learning models could also be investigated to identify subtle semantic and etymological differences between words and improve the accuracy of cognate detection.

Our study focused on the Indo-European language family. To gain a more comprehensive understanding of cognates and language evolution, it would be valuable to extend the analysis to other language families. This would involve collecting datasets of contextually close words from various language families and applying the L2C4 method to uncover cognate clusters. By comparing the clusters across language families, we can gain insights into the historical connections between languages and track the passage of words from parent languages to their descendants.

ACKNOWLEDGEMENTS

Sarthak Rastogi contributes to all parts of this project, including method design, data pre-processing, modelling, result analysis, and paper writing. Manpreet Kaur contributes to result analysis and paper writing.

DISCLOSURE STATEMENT

No potential conflict of interest was reported by the author(s).

REFERENCES

1. T. Rama. Chinese Restaurant process for cognate clustering: A threshold free approach [Internet]. arXiv [cs.CL]. 2016. Available from: http://arxiv.org/abs/1610.06053.

2. G. Mikulyte, and D. Gilbert. An efficient automated data analytics approach to large scale computational comparative linguistics [Internet]. arXiv [cs.CL]. 2020. Available from: http://arxiv.org/abs/2001.11899.

3. R. S. P. Beekes, and M. D. Vaan. *Comparative indo-European linguistics: An introduction*. John Benjamins Publishing, 2011.

4. M. Meier-Brügger, M. Fritz, and M. Mayrhofer. *Indogermanische Sprachwissenschaft, 9., durchgesehene und ergänzte auflage 2010 edition*. Berlin: De Gruyter, 2010.

5. G. Kondrak. Identifying cognates by phonetic and semantic similarity. In: *Second meeting of the north American chapter of the association for computational linguistics on language technologies 2001 - NAACL '01*. Morristown, NJ, USA: Association for Computational Linguistics; 2001.

6. W. Wu, and D. Yarowsky. Creating large-scale multilingual cognate tables. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA); 2018.

7. J. List. LexStat: automatic detection of cognates in multilingual wordlists. 117–125; 2012.

8. O. Frunza, and D. Inkpen, "Identification and disambiguation of cognates, false friends, and partial cognates using machine learning techniques," *Intern. J. Linguist.*, Vol. 1, no. 2, 2010. DOI: 10.5296/ijl.v1i1.309.

9. B. Hauer, and G. Kondrak. 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; 2011.

10. T. Rama, J. Wahle, P. Sofroniev, and G. Jäger. Fast and unsupervised methods for multilingual cognate clustering [Internet]. arXiv [cs.CL]. 2017. Available from: http://arxiv.org/abs/1702.04938.

11. G. Kondrak, and T. Sherif. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In: *Proceedings of the workshop on linguistic distances - LD '06*. Morristown, NJ, USA: Association for Computational Linguistics; 2006.

12. M. S. Wu, and J. M. List. Annotating cognates in phylogenetic studies of Southeast Asian languages. Language Dynamics and Change (published online ahead of print, 2023). DOI: 10.1163/22105832-bja10023.

13. D. Kanojia, P. Sharma, S. Ghodekar, P. Bhattacharyya, G. Haffari, and M. Kulkarni. Cognition-aware cognate detection. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 3281–3292, Online. Association for Computational Linguistics; 2021.

14. J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics; 2014.

15. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space [Internet]. arXiv [cs.CL]. 2013. Available from: http://arxiv.org/abs/1301.3781.

16. D. Kakwani, A. Kunchukuttan, S. Golla, S. Gokul, A. Bhattacharyya, M. M. Khapra, et al., "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Findings of the association for computational linguistics: EMNLP 2020*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 4948–4961.

17. P. A. V. Hall, and G. R. Dowling, "Approximate string matching," *ACM Comput. Surv.*, Vol. 12, no. 4, pp. 381–402, 1980. DOI: 10.1145/356827.356830.

18. L. Philips, "Hanging on the metaphone," *Comput. Lang.*, Vol. 7, no. 12, pp. 39–43, 1990.

19. L. Philips, "The double metaphone search algorithm," *C/C++ Users J.*, Vol. 18, pp. 38–43, 2000.

20. L. J. P. Van Der Maaten, and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, Vol. 9, pp. 2579–605, 2008.

21. B. J. Frey, and D. Dueck, "Clustering by passing messages between data points," *Science*, Vol. 315, no. 5814, pp. 972–6, 2007. DOI: 10.1126/science.1136800.

22. T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *SIGMOD Rec.*, Vol. 25, no. 2, pp. 103–14, 1996. DOI: 10.1145/235968.233324.

23. J. Macqueen. Some methods for classification and analysis of multivariate observations; 1967.

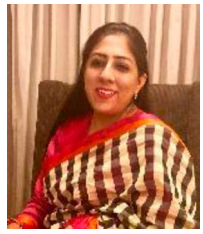
24. W. R. Fox, L. Kaufman, and P. J. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," *J. R. Stat. Soc. Ser. C Appl. Stat.*, Vol. 40, no. 3, pp. 486, 1991. DOI: 10.2307/2347530.

25. K. R. Shahapure, and C. Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 747–748; 2020.

AUTHORS

Sarthak Rastogi is an undergraduate student at Manav Rachna University as well as Indian Institute of Technology, Madras. He is currently engaged in research on the applications of LLMs in Human–Computer Interactions.

Email: thesarthakrastogi@gmail.com.



Manpreet Kaur is serving as a professor in the department of Computer Science and Technology, Manav Rachna University. She has a rich academic experience of working at esteemed institutions for 15 years. Her research interests lie in the areas of natural language processing, information retrieval and machine learning.

Corresponding author. Email: manpreet.kathuria@gmail.com.