Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022), pages 84–93
Language Resources and Evaluation Conference (LREC 2022), Marseille, 25 June 2022

Contextual Unsupervised Clustering of Signs for Ancient Writing Systems

Michele Corazza[1], Fabio Tamburini[1], Miguel Valério[2], Silvia Ferrara[1]
[1]Department of Classical Philology and Italian Studies, University of Bologna.
[2] Departament de Prehistòria, Universitat Autònoma de Barcelona
{michele.corazza2, fabio.tamburini, s.ferrara}@unibo.it
miguel.valerio@uab.cat

Abstract

The application of machine learning techniques to ancient writing systems is a relatively new idea, and it poses interesting challenges for researchers. One particularly challenging aspect is the scarcity of data for these scripts, which contrasts with the large amounts of data usually available when applying neural models to computational linguistics and other fields. For this reason, any method that attempts to work on ancient scripts needs to be *ad-hoc* and consider paleographic aspects, in addition to computational ones. Considering the peculiar characteristics of the script that we used is therefore a crucial part of our work, as any solution needs to consider the particular nature of the writing system that it is applied to. In this work we propose a preliminary evaluation of a novel unsupervised clustering method on the Cypro-Greek syllabary, a writing system from Cyprus. This evaluation shows that our method improves clustering performance using information about the attested sequences of signs in combination with an unsupervised model for images, with the future goal of applying the methodology to undeciphered writing systems from a related and typologically similar script.

Keywords: Deep Learning, ancient writing systems, clustering,inventory of signs in a script.

1. Introduction

The aim of this work is to investigate whether automatic methods can be applied to ancient undeciphered writing systems. One particularly challenging aspect for research can be the sign inventory of a script, as with certain undeciphered scripts there is no consensus among experts. Namely, it can be very difficult to distinguish what is a sign on its own right (grapheme) or a mere variant of a sign (allograph). This issue is detrimental to any attempt at decipherment and it can be further complicated in cases in which the writing system is scarcely attested and the corpus has many damaged inscriptions.

This work constitutes a preliminary investigation of a neural model that aims to learn good latent representations for signs in ancient, undeciphered writing systems. We are interested in the application of computational methods to ancient scripts from the Aegean and Cyprus, in particular to Cypro-Minoan. Cypro-Minoan is a script from the second millennium BCE, attested in Cyprus and the Syrian town of Ugarit. Since there is uncertainty regarding the inventory of signs of this script, we can only use unsupervised methods, which do not use prior information on the status of individual signs. This has the added benefit of avoiding any bias from hypotheses formulated by experts in the field.

In this work, we propose a new method for undeciphered writing systems using images as its input and no gold standard labels. The system improves upon existing methods for images in order to adapt them to this specific domain by incorporating information about the attested sequences of signs. Since no gold standard can be obtained directly from undeciphered writing systems, we describe a preliminary step consisting in the evaluation of our improvement over a baseline, using the Cypro-Greek (CG) syllabary as our ground truth for the evaluation, as CG is descendant script, thus closely related to Cypro-Minoan and it has been deciphered.

2. Related Work

In recent years, the prominence of deep neural networks in natural language processing tasks has increased, leading to improved performance on many tasks. The usage of these models for ancient writing systems however poses unique challenges: these scripts are scarcely attested and when they are undeciphered no evaluation can be performed to assess the performance of neural models. Nevertheless, some scholars have proposed various approaches that deal with ancient writing systems.

In particular, some models tackle the problem of damaged inscriptions, trying to reconstruct textual content in ancient Greek (Assael et al., 2019) and Babylonian Akkadian (Fetaya et al., 2020) using neural models. Another interesting task is the identification of scribal hands, where the goal is to investigate whether documents were inscribed by the same person or not. Computational methods for this task have been applied to the Dead Sea Scrolls (Popović et al., 2021) and to Linear B inscriptions (Srivatsan et al., 2021). Finally, a deep learning model was proposed in order to identify textual content written in the Indus Valley script (Palaniappan and Adhikari, 2017), which constitutes, to the best of our knowledge, the first application of neural networks to an undeciphered writing system.

While in recent years there have been attempts to apply machine learning methods to ancient writing systems, as far as we are aware no unsupervised model has been applied to the inventory of signs of ancient writing systems. Since we are interested in unsupervised

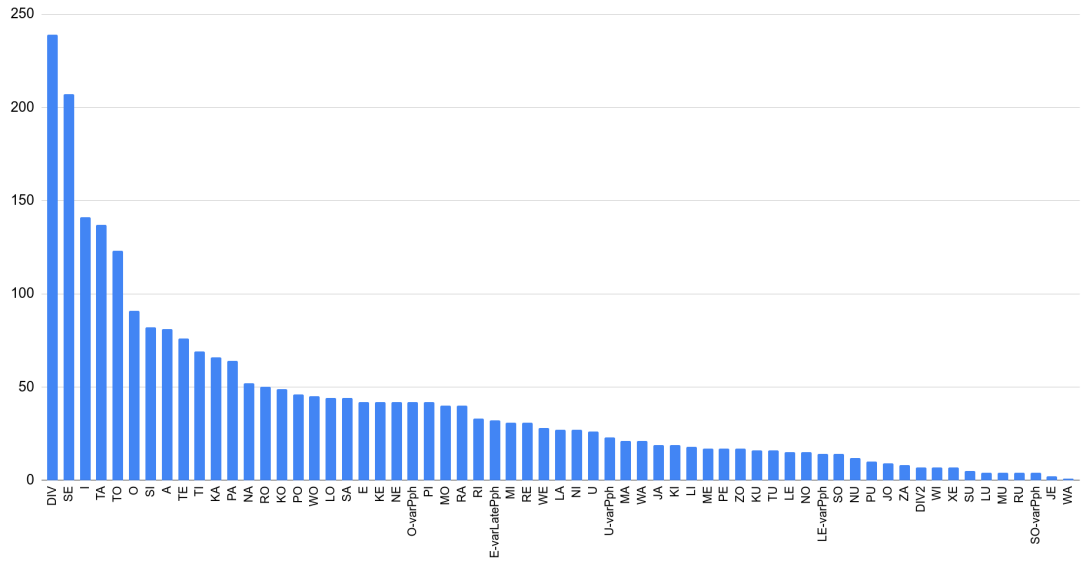

Figure 1: Number of attestations for each sign as represented in our dataset.

approaches, we will now discuss the state of the art of such systems for image classification.

Recent advancements in the application of unsupervised methods to image classification use a multitude of methods, that can be divided in different overarching approaches. Since our evaluation uses clustering as the main task for the model in a two-step approach, we are especially interested in clustering-based models. These are models that use clustering not only after having learned good quality representations for images, but also during training. Some methods using clustering for unsupervised learning on images use Convolutional Neural Networks and perform clustering on the latent representations of images. This is the case, among others, of DEC (Xie et al., 2016), DAC(Chang et al., 2017) and DeepCluster(Caron et al., 2018). Other approaches aim to maximise the mutual information between an image and augmented versions of it. This is the case for IIC (Ji et al., 2019) and IMSAT (Hu et al., 2017). SWAV (Caron et al., 2020) works similarly, by using assignments between two augmented versions of the same image, and using the swapped assignments as the labels to train the model. DeepClusterv2 (Caron et al., 2020) is a combination between SWAV and DeepCluster, using augmented versions of images, but still obtaining pseudo-labels for training from K-Means clustering. SCAN (Van Gansbeke et al., 2020) ditches clustering altogether, and uses a two-step approach: first, it minimizes the distance between an image and its augmentation as a pretext task, then the nearest neighbors of each vector are computed, and used to classify images in the same category.

3. Dataset

To assess the performance of our unsupervised model, we looked for a writing system with three characteristics:

- deciphered status, allowing us to compare our results with a known ground truth;
- a close relationship and typological similarity to Cypro-Minoan, in order to count with signs of the same type (syllabic) and sign inventory of comparable size (some dozens of syllabograms);
- a large enough corpus to provide us with a reasonable amount of data.

The obvious choice was then the Cypro-Greek syllabary, which is the only known script that meets all aforementioned criteria. The script (in use roughly between the 11th or 10th and the 4th centuries BCE) is deciphered and is known to have been adapted from Cypro-Minoan to write a well-understood ancient dialect of Greek (Arcado-Cypriot). Like Cypro-Minoan, its signs are syllabograms that represented open syllables, i.e. Vowel (V) or Consonant-Vowel (CV) syllables. In addition to 56 syllabograms, the Cypro-Greek script also comprised numerical signs and punctuation signs, namely dividers of sequences, which stood for words or groups of words (Egetmeyer, 2010).

Our dataset was obtained from drawings of Cypro-Greek inscriptions from various sources (Casabonne et al., 2002; Egetmeyer, 2010; Masson, 1983; Mitford, 1981; Karageorghis and Karageorghis, 1956; Karageorghis, 1976; Karnava, 2019; Masson and Mitford, 1986; Mitford, 1971; Masson and Olivier, 1983; Mitford, 1958; Mitford and others, 1961; Olivier, 2007; Mitford et al., 1983). The drawings were scanned, and the single signs of each inscription were manually segmented. They were also cropped to obtain square images of 100x100 pixels, retraced as clean black signs on white background. Each file was then labelled with the transcription (reading) of the sign in question. The reading assigned followed reference editions

of the texts (Masson and Olivier, 1983; Egetmeyer, 2010), except for some specific cases where the updated transcription stemmed from individual publications (amongst the ones cited above).

The total number of sign images obtained was 2995 from 164 inscriptions. We then proceeded to exclude images of signs that were broken or damaged, and which therefore did not show their shape in full. Whenever a sign was damaged but the full form was still preserved and drawn, the noise (e.g., cracks or scratches on the inscription medium) was manually removed from the drawing. The number of excluded sign images was 322, so that after this filter we were left with a total of 2673 images.

Because our method considers the context (the position of signs in relation to other signs in the sequences and texts), we gave preference to larger texts written clearly in Greek language. The longest text in the dataset (ICS 217, side B) yielded 584 sign images, while the shortest provided only 2, but on average a document of the dataset provided 9 signs. To make the dataset as representative as possible of the complete corpus of the script, which surpasses 1,050 inscriptions (Egetmeyer, 2010), we deliberately included documents from various geographical areas and different time periods, even if an equal number of signs between locations was not achieved.

The number of categories of signs represented by these images is 64, which includes syllabograms, numerical and punctuation signs, and 'space', which refers to a space in the inscription probably used as a separating device. Importantly, the Cypro-Greek syllabary existed in two main varieties: one used mainly in the area of ancient Paphos, in West Cyprus ('Paphian') and another used in most of the rest of the island ('Common'). The Paphian variety features specific variants of some signs (5 in our dataset), which have different shapes but the same phonetic values as their counterparts in the Common variety. As their shape is significantly different, to the extent where it would affect the clustering method, the images pertaining to these categories received specific labels that distinguished them as Paphian. Finally, out of the 56 syllabograms that make up the sign inventory of Cypro-Greek (excluding the Paphian graphic variants), only one is not represented in our dataset. This is syllabogram XA, as it is a rare sign not found among the 164 inscriptions we compiled.

Like most linguistic features, the sign frequency follows a Zipf distribution (Figure 1), with some categories appearing fewer than 10 times in the entire corpus. This situation, while expected, makes any attempt at creating a neural model classifying signs very challenging, especially since we use an unsupervised method to cluster them. The most common grapheme is the divider denoted in the plot by "DIV". This sign is used to separate sign sequences, which in the Cypro-Greek script can stand for single words or entire phrases, such as 'the city of Idalion'.

4. Model

As the basis for our approach we use DeepClusterv2 (Caron et al., 2020), an unsupervised convolutional model for images, an improvement on the original DeepCluster (Caron et al., 2018) algorithm. DeepCluster (Figure 2) is an unsupervised model that applies K-Means to the output of a convolutional neural network, a ResNet50 (He et al., 2016), in order to learn pseudo-labels that are then, in turn, used to update the weights of the model. Before each epoch the vectors representing all of the signs are obtained from the model. These are then normalized to be unit vectors by dividing them by their L2 norm. On these, a K-means clustering algorithm is applied, obtaining pseudo-labels that can be used to train the model on a classification task.

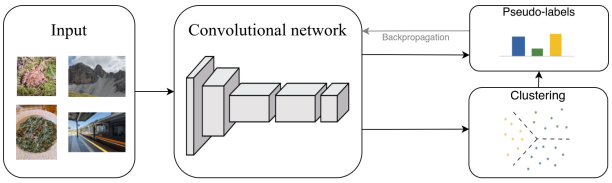

Figure 2: DeepCluster

DeepClusterv2 improves upon its predecessor in some significant ways:

- It replaces the output layer of DeepCluster with one obtained by using the centroids of the clusters from K-means. The application of this output layer to the vectors from the ResNet50 calculates the dot product between each vector and each centroid. Since both the centroids and the vectors representing images are normalized unit-norm vectors, this corresponds to the cosine similarity between vectors and centroids. With this method, the output layer does not need to be reinitialized after every epoch and the proximity of the sign to its centroid is enforced directly in the model;
- The model uses random augmentations of the images (crops, color distortion, random flips) both before clustering and when training the model;
- Other minor adjustments include cosine learning rate and the usage of a multi-layer perceptron as a projection head for the image vectors.

Our model, Sign2Vec$_c$ (Figure 3), improves upon the existing DeepClusterv2 approach by considering the role of contextual information when dealing with images representing signs. In fact, the preceding and following sign bear important information when attempting to detect allographs in writing systems, as similar sign shapes found within the same position of a sequence are more likely to be variants of the same sign. This information is often used by paleographers, as it can give precious insight into the allography of signs and it is also a crucial aspect for any attempt at decipherment.

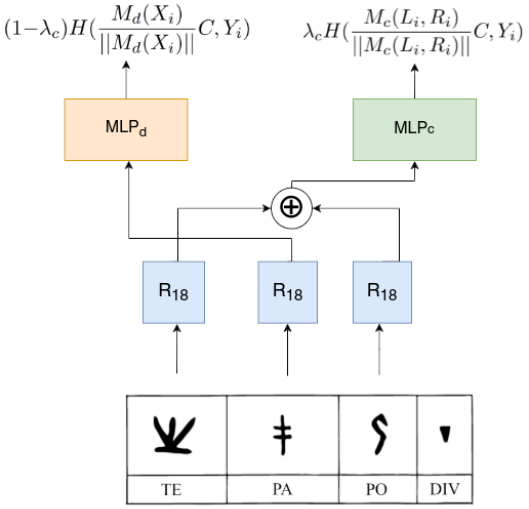

Figure 3: Sign2Vec$_c$.

Sign2Vec$_c$ is inspired by the CBOW approach (Mikolov et al., 2013) often used in computational linguistics to learn word embeddings. In this approach, a word is predicted from its left and right context. Similarly, our aim is to train a model that can predict a sign from its context. In our case, however, we have no source of supervision and cannot provide the model with a symbolic representation of signs, since images are our only input. Additionally, we do not have labels that can be used to directly train a model to predict a syllabogram from its context. Therefore, we extend the DeepClusterv2 framework by using a joint learning objective. In addition to the usual DeepClusterv2 loss, we use the signs to the left and right of the one under examination in order to predict the cluster that the central sign belongs to. The choice of such a small context window (size one) might seem low when compared to larger context windows traditionally used in computational linguistics. However, its properties fit our task well: as CG is a syllabary, by limiting the window size to one, we never cross the boundaries of syllabic sequences, as there is always a sign separating them. Also, with larger context windows, when dealing with a sign found at the end of a document, we would need to introduce extra virtual signs on the right side (and the same applies to the left side at the beginning of a text), which is problematic.

Formally, we use the following training objective:

$$\mathcal{L}(\mathrm{C}, \mathrm{X_i}, \mathrm{Y_i}) = (1-\lambda_c)\ H(\frac{M_d(X_i)}{||M_d(X_i)||}C, Y_i) + \lambda_c\ H(\frac{M_c(L_i,R_i)}{||M_c(L_i,R_i)||}C, Y_i) \quad (1)$$

Where $C \in \mathbb{R}^{v_s \times n_c}$ is a matrix representing all the centroids of the clusters obtained from K-means, and v_s is the size of the vectors obtained from the model, while n_c is the number of clusters obtained from K-Means. X_i is the central sign, while L_i, R_i represent the signs to the left and right of X_i, respectively. Y_i is the cluster that the central sign X_i belongs to according to K-means. H is the categorical cross entropy. Consider the fact that since we normalize both branches of the loss by dividing the vectors by their L2 norm, they both have unit norm. Therefore the product between the vectors obtained from the model and C corresponds to the cosine similarity of the vector with each centroid. λ_c is a constant used to determine the relative weight of the two components of the loss.

The neural model is implemented by the two functions M_d and M_c:

$$M_d(X_i) = MLP_d(R_{18}(X_i))$$
$$M_c(L_i, R_i) = MLP_c(R_{18}(L_i) \oplus R_{18}(R_i))$$

Where MLP_d and MLP_c are two multi-layer perceptrons that project the central sign and the concatenation ($\oplus$) of the left and right sign respectively to a vector of the same size. The outputs of M_d and M_c are both in $\mathbb{R}^{b_s \times v_s}$, where b_s is the batch size. Notice that, therefore, the matrix products of equation 1 between MLP_d, MLP_c respectively and C are in $\mathbb{R}^{b_s \times c_n}$, so they calculate, for each image in a mini-batch, a proximity to all centroids. As MLP_d and MLP_c operate on vectors with different sizes and perform different tasks, they do not share weights. R_{18} is the ResNet18 convolutional network that we use to replace the ResNet50 present in the original implementation of DeepClusterv2 to reduce the number of parameters. It is shared by both branches of the model.

Since Sign2Vec$_c$ uses contextual information to improve the base DeepClusterv2 model, there are some peculiar situations that arise. First, we need to consider how to provide context to the model at the beginning or end of inscriptions. For this situation, we can leverage a peculiar characteristic of the Cypro-Greek syllabary, which is also present in the Cypro-Minoan script: the system uses vertical lines or dots as sequence separators, so we can simply use a random sequence divider from the corpus to replace the beginning or the end of an inscription in the context, since the limits of a document also represent sequence boundaries. This random sequence divider is chosen at run-time and altered at every epoch for a given context, since always choosing the same separator from the dataset would be arbitrary. This also means that we implicitly provide the model with some information about separators. However, dividers are not syllabograms and do not encode phonetic information, so we can safely exclude them from any further evaluation. Additionally, since they are very frequent, specialists agree on their function even in the undeciphered Cypro-Minoan script and they can be distinguished from other signs without any uncertainty.

Another aspect that needs careful consideration is the fact that some signs are damaged and some inscriptions are broken. In this case, when we needed to represent a damaged sign or a broken portion of the inscription, we generate random black dots on a white

background at run time, using Poisson disc sampling (Bridson, 2007). This choice was made in an effort to reduce the effect that a fixed image representing damage would have on the model, since this might lead the model to rely on the fixed "damage" image, while the missing signs that are damaged are variable in nature. The usage of dots matches the conventional representation of damage used by some paleographers in their drawings.

5. Experimental settings

In this section, we provide additional information on the settings and hyper-parameters we use to train all our models. The first important aspect to consider regards the parameters used in order to obtain the augmented versions of images during training. In our models, we use two sets of cropped augmentations for each image, with a relative size compared to the original image chosen randomly in the ranges $[0.6, 1.0]$, $[0.4, 0.6]$. The two sets of crops are 6 and 10 for each image, respectively.

We did not alter the rest of the augmentation steps used by DeepCluster, which include a random horizontal flip of the image and a random color distortion. It needs to be noted, though, that while it is sensible for the Cypro-Greek syllabary, the application of a random horizontal flip might not be suitable in general, as it introduces an invariance with respect to flipped images that might be problematic. Since, however, the Cypro-Greek syllabary doesn't contain distinct graphemes that are horizontally flipped, we conclude that there is no reason to drop this augmentation step. The only alteration we made to the original augmentation is the reduction of the strength of the color distortion by using a parameter of 0.1 instead of the default 1.0, considering that we worked on black and white images and that such a strong level of color distortion was making the signs barely distinguishable from the background. We provide the values for all hyper-parameters used to train the model in Table 1.

Another important aspect of our evaluation is the choice of the number of clusters provided to K-Means (number of prototypes in Table 1). Since we are interested in evaluating the performance of our model by simulating its application to an undeciphered writing system, we cannot provide the model with the exact number of signs present in the dataset. We therefore proceed by overclustering the signs, and use a very generous estimate of 100 which should be more than any kind of system based on the syllabograms it contains. The fact that 100 is repeated three times in the parameters means that we apply K-means clustering three times. Naturally, we also have three different output layers for the model, one for each K-Means application. Since the algorithm initializes centroids at random, running K-means multiple times increases the robustness of the model and reduces the impact of the random initialization of the centroids.

Hyper-parameter	Value
Architecture	Resnet18
Base Learning Rate	4.8
Batch size (b_s)	16
Crops for assign	0
Epochs	100
Feature dimensions (v_s)	128
Final learning rate	0.0048
Number iterations before freeze	300000
λ_c	0.2
Hidden MLP size	2048
Max scale crops	[1.0, 0.6]
Min scale crops	[0.6, 0.4]
Number of crops	[6, 10]
Number of prototypes (n_c)	[100,100,100]
Size of the crops	[80, 60]
Start warmup	0.3
Temperature	0.1
Warmup Epochs	10
Weight decay	1×10^{-6}

Table 1: Hyper-parameters for Sign2Vec$_c$

Since we cannot use the number of classes during training, K-Means, which needs this information to initialize its centroids, can't be used as a clustering algorithm to evaluate performance. We are also unable to use the output layer of the model directly, since it overclusters our data. We therefore use a density based clustering algorithm, DBSCAN (Ester et al., 1996), which does not require the number of clusters as an input, in order to evaluate the performance of our model. The algorithm is applied to the latent representations of single signs learned by the models, given by:

$$\frac{M_d(X_i)}{||M_d(X_i)||}$$

We use the implementation of DBSCAN from scikit-learn (Pedregosa et al., 2011).

6. Results and evaluation

To evaluate the effectiveness of Sign2Vec$_c$ on the CG dataset, we need to perform a comparison with a DeepClusterv2 model trained with the same parameters but no context. However, we also need to adapt the model so that DBSCAN can be applied. In particular, we note that using oversampling is the best way to increase the density of signs belonging to the same class, allowing the usage of DBSCAN as a clustering algorithm. However, oversampling minority classes is not possible as we have no access to the ground truth labels. For this reason, we apply oversampling by replicating the entire dataset twice. This approach allows us to obtain two objectives: on one hand, we keep the centroids fixed for a longer time, since every epoch is twice the length of a standard one. On the other, we also oversample less frequent signs when applying K-Means, thus helping the clustering algorithm to detect more rare shapes and create a cluster around them. It is worth noting that,

Model	ϵ value	Adjusted Rand Index	Adjusted Mutual Information	V-measure
DC2, no oversample	0.05	0.30 ± 0.02	0.59 ± 0.02	0.66 ± 0.03
DC2, oversample	0.06	0.47 ± 0.02	0.68 ± 0.02	0.73 ± 0.01
S2V, oversample	0.08	$\mathbf{0.51 \pm 0.04}$	$\mathbf{0.72 \pm 0.02}$	$\mathbf{0.75 \pm 0.01}$

Table 2: Means and standard deviations for all the best clustering metrics of the three models.

Models	Adjusted Rand Index	Adjusted Mutual Information	V-measure
DC2 with and without oversampling	$1.92 * 10^{-5}$	$1.21 * 10^{-4}$	$1.71 * 10^{-4}$
S2V with oversampling, DC2 with oversampling	0.01	$3.60 * 10^{-5}$	$1.17 * 10^{-4}$

Table 3: One tailed t-tests comparing the metrics obtained from the models.

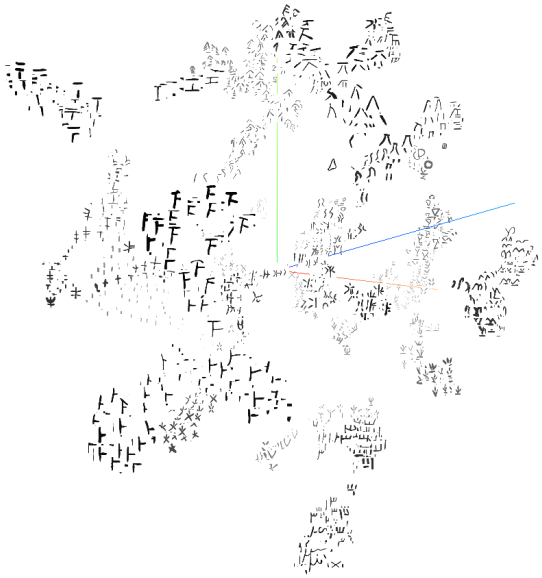

Figure 4: Three dimensional t-SNE projections for sign representations.

since every epoch and every sign is subject to random crops, the two copies of the same sign are not identical and therefore this method of oversampling has a positive effect on the application of K-means as well.

Before quantitatively evaluating performance using DBSCAN, another useful output of the model is the possibility to create three-dimensional scatter plots from the sign representations (Figure 4), in order to visualize the distance between signs. Since both DeepClusterv2 and Sign2Vec$_c$ work by minimizing distances between similar signs, the best choice for a dimensionality reduction algorithm is to use t-distributed stochastic neighbor embedding (t-SNE), which uses the Kullback-Leibler divergence between the distributions of distances in the original space and those in the reduced space (Van der Maaten and Hinton, 2008). By applying t-SNE (from scikit-learn) to the outputs of all three models we can create three visualizations[1] of the vector space which can be used by experts to spot incorrectly classified signs. To a lesser extent, we can also qualitatively assess the improvement that we obtain by applying Sign2Vec$_c$ and we see that, in general, Sign2Vec$_c$ tends to create groups of signs that are more separated from each other when compared to DeepClusterv2. This is especially evident when we compare the scatter plot from Sign2Vec$_c$ with the one obtained from DeepClusterv2 and no oversampling, as those have the largest difference in terms of performance. However, evaluating performance on the scatter plot alone is unfeasible, as the data is highly multidimensional and it is not always clear which model performs best. Scatter plots are not just useful for coarse evaluations of models. They also make for a state-of-the-art visual tool with important applications and implications for the paleographic study of ancient scripts. They can provide specialists with a method for quickly comparing large numbers of sign shapes, and, in that way, independently postulate hypotheses about the classification of graphemes or even identify misread signs.

We show the improvement in performance when using overclustering with DeepClusterv2, then we evaluate the further improvement in performance obtained by Sign2Vec$_c$. In order to compare models, we retrain each of them 10 times, in order to reduce the impact of the random initialization of the parameters as a factor and test for the statistical significance of the results.

Since we already use sequence dividers as a given to replace the end of sequences, we exclude them from the evaluation of clusters. In the same way, we exclude numerals from the evaluation, as they are not syllabograms and hence not our main focus. Moreover, the basics of the system for writing integers is largely shared by all related Aegean and Cypriot scripts (Linear A, Linear B, Cypro-Minoan, and Cypro-Greek).

When applying DBSCAN for our numerical evaluation, however, two parameters must be established. The first one is the minimum number of neighbors needed for a point to be considered a core point in the algorithm. Since we are using an unsupervised approach, we cannot assume any minimum size for these local neighborhoods, so we choose the minimum possible value of 2. Another crucial parameter required by DBSCAN is an ϵ value that controls the maximum distance

[1] https://corpora.ficlit.unibo.it/INSCRIBE/PaperCG/

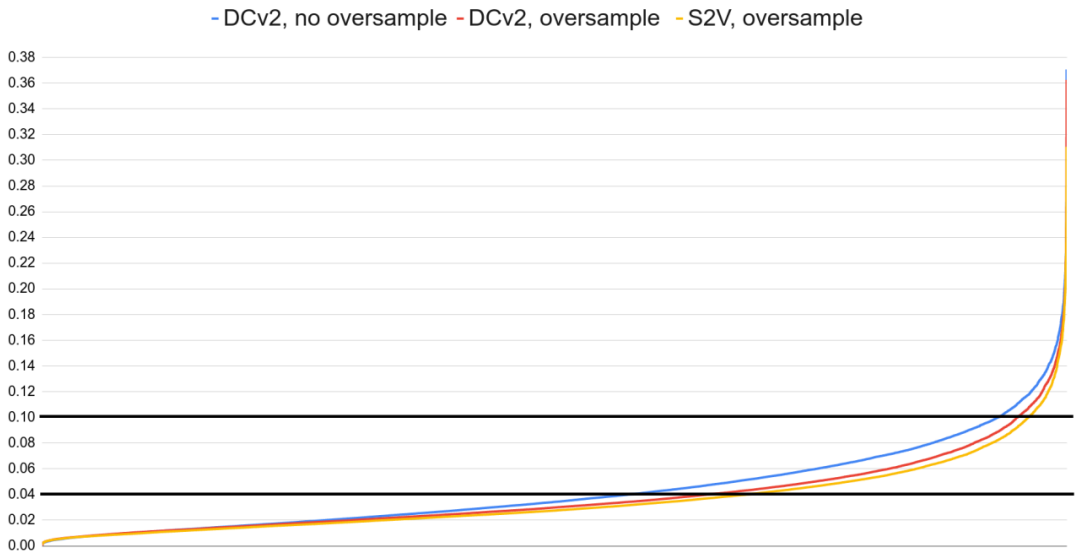

Figure 5: The elbow plots obtained from DeepClusterv2 with no oversampling, DeepClusterv2 with oversampling and Sign2Vec$_c$ with oversampling. The two horizontal lines show the range of ϵ values we used for our evaluation.

between a vector and its neighbors to initialize the algorithm. This parameter indirectly controls the number of clusters that will be created, as well as the number of signs that are deemed to be impossible to cluster by the algorithm. Lower values of ϵ result in a higher number of clusters, while higher values create fewer clusters. One of the few proposals for a heuristic to choose ϵ is the elbow criterion (Rahmah and Sitanggang, 2016). This method works on a vector space by first computing, for each vector, the farthest amongst its two nearest neighbors. Then, these values are sorted in ascending order and the elbow obtained by using this method is used to select the value of ϵ. This corresponds to finding a point of diminishing returns, where increasing ϵ does not result in many more vectors having local neighbors.

In Figure 5 we show the elbow plot obtained by sorting the maximum distance from the two nearest neighbors of each sign. While this criterion is useful, in practice we notice that when applied to approximately 27000 signs (the total number of signs x 10 models), the elbow can be ambiguous and does not always lead to an acceptable level of performance for all models. Moreover, we will show that Sign2Vec$_c$ with oversampling appears to tolerate a wider range of values for ϵ, while this is not true for the non-contextual versions of the model. While Sign2Vec$_c$ is superior to DeepClusterv2 in this aspect, we consider an arbitrary choice of ϵ as unfairly advantageous to our model. Therefore, we choose to evaluate the relative performance of the three models over a range of ϵ values, also shown with black lines in Figure 5. While it is debatable where the elbows lie in this kind of figure, we use a wide range to avoid the reliance on a single value of ϵ. Even if it can be argued that we do not include the elbow for all models, the results show that we do consider the best performing values of ϵ for all of them.

To assess the clustering performance of all three models we use some standard metrics for clustering: Adjusted Rand score, Adjusted mutual information and V-Measure, as implemented by scikit-learn (Pedregosa et al., 2011). As we use a range of values of ϵ for our evaluation, we provide two different ways to show the improvement in performance obtained from Sign2Vec$_c$: we plot all the mean values of the metrics for the different values of ϵ, then we select the best value for each model and compare them using a one tailed t-test to evaluate the statistical significance of the observed difference in performance between models.

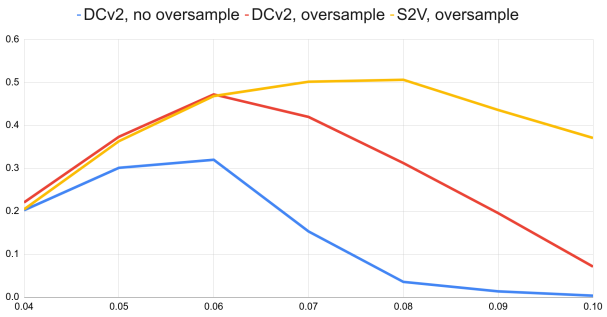

Figure 6: Adjusted Rand score of the three models for different values of ϵ.

By observing the mean of each metric obtained from the models with varying ϵ values (Figures 6,7,8), we can clearly spot some interesting trends. First, we consider a wide enough range of values for ϵ that the global maximum for all metrics is included, while at the edges of the plot we observe decreasing performance. When comparing the oversampled variant of DeepClusterv2 to the non oversampled one, we can see a marked improvement across all metrics, suggesting that over-

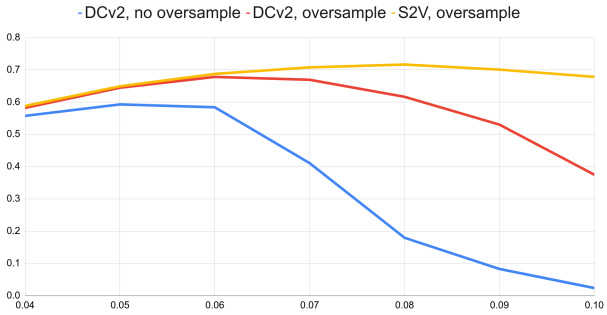

Figure 7: Adjusted mutual information of the three models for different values of ϵ

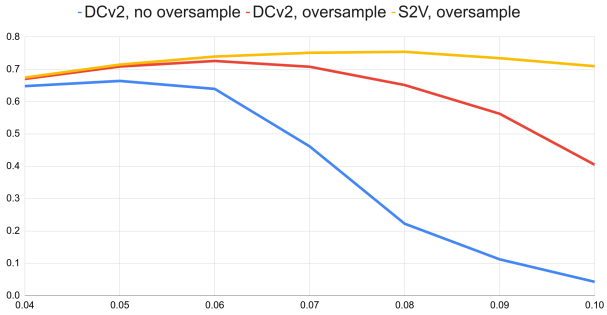

Figure 8: V-measure of the three models for different values of ϵ.

sampling leads to a measurable improvement in performance when using DBSCAN for clustering. Additionally, our Sign2Vec$_c$ model achieves the highest values. While for low values of ϵ Sign2Vec$_c$ and the oversampled version of DeepClusterv2 show similar performance, for $\epsilon > 0.06$ Sign2Vec$_c$ performance clearly improves while DeepClusterv2 shows a sharp decrease across all metrics. Also, Sign2Vec$_c$ appears to be more stable than DeepClusterv2 across a wider range of ϵ values. In practice, this means that Sign2Vec$_c$ is preferable for any attempt at automatic clustering on an undeciphered script, where the number of clusters is not known in advance and ϵ can only be chosen by using heuristics such as the elbow method or by evaluating the quality of the clusters manually.

Table 2 shows the means and standard deviations across all metrics for the best performing values of ϵ of each model. The metrics show a clear trend that reflects the improvement of the oversampled model with respect to the non oversampled variant, while the best performing model overall is Sign2Vec$_c$. Table 3 presents the results of the t-tests comparing the metrics obtained by the various models. We compare DeepClusterv2 with and without oversampling, DeepClusterv2 with oversampling and Sign2Vec$_c$, respectively. The table shows that all differences observed in the metrics are in fact statistically significant ($p < 0.05$) even with a relatively small sample size of 10 models. This, in conjunction with the aforementioned advantage of Sign2Vec$_c$ even when considering multiple values of ϵ, shows that using context in order to augment the vector representations obtained from DeepClusterv2 leads to improved clustering performance that cannot be due to random chance.

7. Conclusions

In the previous sections, we describe the peculiar challenges that are associated with the application of machine learning models to ancient writing systems, with particular attention to undeciphered scripts. In particular, we focus on syllabic systems from the Aegean and chose the Cypro-Greek syllabary as our gold standard, in order to be able to create an ad-hoc system that deals with such scripts.

We then propose an evaluation framework that can be used to assess whether performance improvements over existing methods can be obtained by tailoring the approach to ancient scripts. In particular, this approach uses DBSCAN as a clustering algorithm over the sign representations learned from neural models, since it allows us to obtain clusters without directly providing their exact number to the system, since this value might be unknown in the context of undeciphered scripts. Furthermore, we use contextual information in an unsupervised model for undeciphered scripts called Sign2Vec$_c$, and prove that this model leads to a clear improvement in performance over the baseline.

The evaluation of the different models on the Cypro-Greek syllabary shows two interesting findings. We observe that using oversampling can be useful when data is scarce, as it greatly improves performance while clustering using DBSCAN. In addition to that, we show that including contextual information leads to a further improvement in performance, suggesting that the usage of context helps the model to generalize variations in shape of the same sign, by also considering its position in sequences. This last finding matches the common approach used by experts, that evaluate the status of signs by examining their position in sequences. This work constitutes, to the best of our knowledge, the first application of unsupervised methods to the sign inventory of ancient writing systems, with the goal of a future application of a similar approach to undeciphered scripts. In addition, it is the first method integrating contextual information with an unsupervised neural model that directly uses the graphical representations of signs.

Acknowledgments

The research contained in this article is part of the ERC Project "INSCRIBE. Invention of Scripts and Their Beginnings". This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 771127).

8. Bibliographical References

Assael, Y., Sommerschield, T., and Prag, J. (2019). Restoring ancient text using deep learning: a case

study on greek epigraphy. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6367–6374. Association for Computational Linguistics.

Bridson, R. (2007). Fast poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches*, 10(1):1.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, pages 139–156.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 9912–9924.

Casabonne, O., , and Egetmeyer, M. (2002). À propos du sceau de diweiphilos. In *Notes ciliciennes*, volume 10 of *Anatolia Antiqua*, pages 177–181. Institut français des études anatoliennes.

Chang, J., Wang, L., Meng, G., Xiang, S., and Pan, C. (2017). Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887.

Egetmeyer, M. (2010). *Le dialecte grec ancien de Chypre. Tome I: Grammaire; Tome II: Répertoire des inscriptions en syllabaire chypro-grec.* De Gruyter.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Fetaya, E., Lifshitz, Y., Aaron, E., and Gordin, S. (2020). Restoration of fragmentary babylonian texts using recurrent neural networks. *Proceedings of the National Academy of Sciences*, 117(37):22743–22751.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hu, W., Miyato, T., Tokui, S., Matsumoto, E., and Sugiyama, M. (2017). Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR.

Ji, X., Henriques, J. F., and Vedaldi, A. (2019). Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874.

Karageorghis, V. and Karageorghis, J. V. (1956). Some inscribed iron-age vases from cyprus. *American Journal of Archaeology*, 60(4):351–359.

Karageorghis, J. (1976). Une cruche chypriote inscrite du début du 5e siècle av. notre ère. *Studi Ciprioti e rapporti di scavo*, 2:59–68.

Karnava, A. (2019). Old inscriptions, new readings: A god for the rantidi sanctuary in south-west cyprus. *Cahiers du Centre d'Études Chypriotes*, 49:19–36.

Masson, O. and Mitford, T. B. (1986). Les inscriptions syllabiques de kouklia-paphos.

Masson, É. and Olivier, M. (1983). Appendix 4: Les objets inscrits de palaepaphos-skales. In V. Karageorghis et al., editors, *Palaepaphos-Skales: An Iron Age Cemetery in Cyprus*, Ausgrabungen in Alt-Paphos auf Cypern, pages 411–415. Universitätsverl.

Masson, O. (1983). Les inscriptions chypriotes syllabiques: recueil critique et commenté. *(Étude chypriotes 1). Réimpression augmentée*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mitford, T. B. et al. (1961). Unpublished syllabic inscriptions of the cyprus museum. *Minos*, 7:15–48.

Mitford, T., Masson, O., and Institut, D. A. (1983). *The Syllabic Inscriptions of Rantidi-Paphos*. Ausgrabungen in Alt-Paphos auf Cypern. Universitätsverlag Konstanz.

Mitford, T. B. (1958). Three inscriptions of marium. *Bulletin of the Institute of Classical Studies*, 5:58–60.

Mitford, T. B. (1971). *The inscriptions of Kourion*, volume 83. American Philosophical Society.

Mitford, T. B. (1981). *The Nymphaeum of Kafizin: the inscribed pottery*, volume 2. De Gruyter.

Olivier, J.-P. (2007). *Édition holistique des textes chypro-minoens*. Fabrizio Serra Editore.

Palaniappan, S. and Adhikari, R. (2017). Deep learning the indus script.

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

Popović, M., Dhali, M. A., and Schomaker, L. (2021). Artificial intelligence based writer identification generates new evidence for the unknown scribes of the dead sea scrolls exemplified by the great isaiah scroll (1qisaa). *PloS one*, 16(4):e0249769.

Rahmah, N. and Sitanggang, I. S. (2016). Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. In *IOP conference series: earth and environmental science*, volume 31, page 012012. IOP Publishing.

Srivatsan, N., Vega, J., Skelton, C., and Berg-Kirkpatrick, T. (2021). Neural representation learning for scribal hands of linear b. In *ICDAR 2021 Workshop on Computational Paleography*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., and Van Gool, L. (2020). Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer.

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.