Self-Supervised Deep Metric Learning for ancient papyrus fragments retrieval

Antoine Pirrone, Marie Beurton-Aimar, Nicholas Journet

IJDAR manuscript No.
(will be inserted by the editor)

Self-Supervised Deep Metric Learning for ancient papyrus fragments retrieval

Antoine Pirrone · Marie Beurton-Aimar · Nicholas Journet

Received: date / Accepted: date

Abstract This work focuses on document fragments association using Deep Metric Learning methods. More precisely, we are interested in ancient papyri fragments that need to be reconstructed prior to their analysis by papyrologists. This is a challenging task to automatize using machine learning algorithms because labeled data is rare, often incomplete, imbalanced and of inconsistent conservation states. However, there is a real need for such software in the papyrology community as the process of reconstructing the papyri by hand is extremely time consuming and tedious. In this paper, we explore ways in which papyrologists can obtain useful matching suggestion on new data using *Deep Convolutional Siamese-Networks*. We emphasize on low-to-no human intervention for annotating images. We show that the *from-scratch self-supervised* approach we propose is more effective than using knowledge transfer from a large dataset, the former achieving a *top-1* accuracy score of 0.73 on a retrieval task involving 800 fragments.

Keywords Ancient Document Reconstruction · Deep Metric Learning · Information Retrieval · Self-supervised Learning · Domain adaptation

Fig. 1 Example of fragments of papyri (source : the GESHAEM Project, `geshaem.huma-num.fr`)

1 Introduction and context

Through the study of ancient documents, archaeologists want to understand how ancient human societies were organized. They want to get a better understanding of the cultural, administrative, societal and economical aspects of the day to day lives of the people living at the time the documents were redacted. When very ancient documents are found, they often have been the subject

The research leading to this results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 758907 and is part of the *GESHAEM* project, hosted by the *Ausonius* Institute. The source code (upon request) and data used in this article is available at `https://morphoboid.labri.fr/self-supervised-papyrus.html`

Antoine Pirrone
E-mail: antoine.pirrone@labri.fr

Marie Beurton-Aimar
E-mail: beurton@labri.fr

Nicholas Journet
E-mail: journet@labri.fr

of important degradation due to their storage conditions and the passing of time. Documents can be torn into multiple pieces, some pieces might be missing and the text can sometime be nearly erased (see Fig. 1).

Studying the content of these documents requires a reconstruction step of the documents from their fragments, without knowing which fragments come together as a single document. This is analogous to having a bag of puzzles in which all the pieces are mixed together and without knowing if all the pieces are present. When there are thousands of fragments which likely constitutes several hundreds of documents, the human time needed to perform the reconstruction step is enormous and the process is extremely tedious.

With the improvements of machine learning methods over the last decades, it has become possible to design software approaches to help solving the task of automatic fragment association. Several specificities of the data we are working with make this process very difficult, such as the fact that fragments coming from the same papyrus may sometime look similar, and sometime look different. The same goes for fragments that do not come from the same papyrus. They are similar in the sense that they are all documents, but the textures, colors and shapes of the fragments can be very different, as well as the sizes, colors and styles of the writings which can vary a lot. Moreover, two fragments originating from the same document can look different because of different conservation conditions (see Fig. 2). On the contrary, two fragments originating from different papyri can look the same because they were written close together in time by the same person with the same material, and conserved in similar conditions (see Fig. 3).

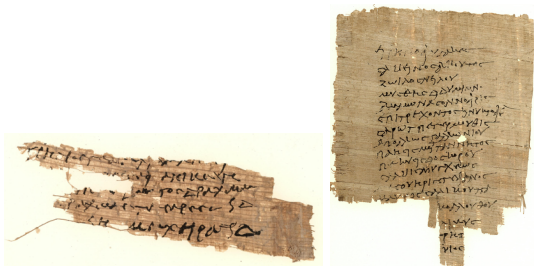

Fig. 2 Example of two fragments that belong to the same papyrus, but look different (with respect to color, texture, papyrus fiber and writing style)

One can see that trying to design hand crafted methods to classify these fragments would be very difficult. Discriminative features must be very fine in order not to be tricked by lots of subtle similar looking but dissimilar examples and vice versa. Today, Deep Learning methods provide a very powerful medium for automatically discovering fine discriminative features with respect to the data available.

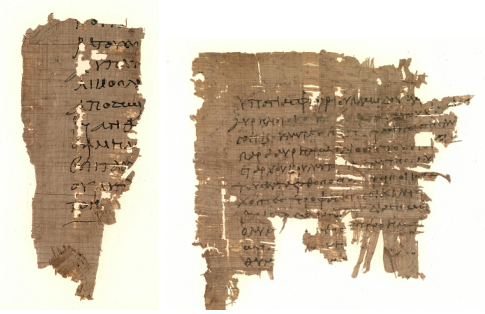

Fig. 3 Example of two fragments that belong to different papyri, but look similar (with respect to color, texture, papyrus fiber and writing style)

As we don't know in advance the number of documents (classes) that we will have to classify the fragments in, we cannot use standard Deep Learning classification schemes. We chose to work with Deep Metric Learning (*DML*) methods [18], which are Deep learning methods for learning similarity between inputs of arbitrary data. In substance, we want to train a *DML* model so that it outputs a large score between two fragments originating from the same document, and a low score for two fragments originating from different documents.

Our ideal goal would be to learn a general *DML* model capable of correctly associating fragments on any data. Unfortunately, this would require an enormous amount of very diverse annotated data that is simply not available. However, with sufficient training data we can train a *DML* that performs well on a given dataset. Using *transfer learning*, or more precisely *domain adaptation*, the information learned on this dataset can be used on another dataset with insufficient training data on its own. This approach still implies that some annotations are available on the target dataset to perform the *fine-tuning*, and since we want as little human annotation work as possible, we also explore the idea of *self-supervised* learning.

The experiments carried out in this paper are made on two datasets. We use the HisFrag database [40], a very large and annotated semi-synthetic document fragments database as the training base, and a papyrus database created from a subset of the Papyrus Collection of the University of Michigan [1] that we created. The paper is organized as follows :

In section 2, we give an overview of existing works that use *DML*, *transfer learning* and *self-supervised* ap-

[1] Found here : https://quod.lib.umich.edu/a/apis (accessed October 28, 2020)

proaches in the context of historical documents information retrieval. Section 3 describes in details the two datasets, and how we created the Michigan dataset. In section 4, we present our training and evaluation protocols. We then present in section 5 the baseline results with and without *domain-adaptation*. Finally, we present in section 6 our *self-supervised* learning approach.

In summary, our main contributions are the following :

1. We provide a challenging and sizeable papyrus fragments dataset containing 4579 fragments constituting 1118 papyri, tailored for fragment retrieval tasks. It is based on the University of Michigan Papyrus Collection, pre-processed and ready to use.
2. We propose *self-supervised Deep Metric Learning* method able to provide useful suggestions of fragment association to papyrologists, that does not need any manual annotation work.
   - We evaluate the proposition on two datasets, with two convolutional neural networks architectures
   - We compare our *self-supervised* approach with a *domain adaptation* approach
   - We provide insight on how this could be useful for papyrologists in a realistic use case

2 Related Works

Information retrieval in the context of historical handwritten or printed documents has recently received a lot of attention by the document image processing community. For instance, a number of competitions were launched with interests in writer identification, page retrieval, content classification and more [40] [10] [6] [5]. Along with these competitions, new datasets are published, which constitute an invaluable source of high quality data on which researchers of the community can compare their methods with each others. This growing interest has led to the publication of numerous and diverse approaches for solving these tasks. While non *Deep Learning* based methods are still proposed [5], there is a growing trend of using the power of *Deep Learning* based methods, as in many other computer science fields [24]. Even with the growing amount of annotated data available in this field, there is most of the time not enough data to properly train *Deep Learning* models with the goal of *generalization*. Meaning obtaining a single model that performs well on different datasets and different types of documents. Thus, the community turns to solutions employing *transfer-learning* or *domain-adaptation*, such as in [43] and [41], or employing *unsupervised* [4], *semi-supervised* [17] or *self-supervised* learning [33] [30] [28].

When trying to perform information retrieval, the specific sub-domain that is used most of the time is *Deep Metric learning* (*DML*) [18]. The idea of this concept is to learn a distance metric given specific data. The data specifies what samples are "close" (or similar) or "far away" (or dissimilar) from each other. This can be implicitly deduced from the labels, two samples that have the same label can be considered as "close". The *DML* model then learns what features are relevant to extract, and learns to project these features in a latent space in which similar samples are close together in the sense of a geometrical distance metric, such as the Euclidian distance and vice versa. This approach has the advantage that it does not require the knowledge of the number of classes in advance to train as in more "traditional" Deep Learning classification methods. Indeed, the only labels needed are the similarity for each given pairs.

Siamese Neural Networks are *Deep Metric Learning* approaches that have been originally introduced by [2] for signature verification. They are a special kind of neural network architecture in which there are two identical branches with shared weights during training and prediction. Each branch outputs a feature vector (embeddings) when given an input. Works in the domain of ancient document making use of *Siamese Networks* include [44], which works on ancient manuscript documents, [31] which works with ostraca fragments, an unusual support, and [36], working on ancient papyrus fragments. More recently, [13] introduced *Triplet Networks*, a variant of the *Siamese Networks*. Instead of using similar and dissimilar pairs, the network is given a triplet with an Anchor sample, a Positive sample and a Negative sample. Optimizing a loss function based on both the distance between Anchor/Positive and Anchor/Negative every time the loss is computed is expected to improve intra-class compactness and inter-class separability [18] compared to losses in which the similar and dissimilar examples are considered separately. Triplet mining strategies such as *batch-hard and batch-all* have also been introduced along with the triplet losses in order to optimize the learning process by selecting triplets that are most likely to significantly contribute to the convergence at a given training step. Different loss functions on triplets have been proposed, but the most widely used is the Triplet loss as defined in [39].

The first appearance of the notion of *self-supervised learning* is from [34], where the authors describe a process for recognizing vowel sounds using a *self-supervised*

learning method. It is usually considered as a subset of *unsupervised learning*. In recent years, there has been many proposals using this approach for feature learning in a broad sense [26], [32], [37] [16]. When working in a *self-supervised* framework, we train the models using a *pretext-task* that can be trained from the unlabeled data itself. The *pretext-task* must be designed in such a way that forces the models to learn useful representations that relate to the task we actually want to achieve, the *target-task* (sometimes called *downstream-task*). Examples of *pretext-tasks* are recovering color from grayscale images [46], [22] and recovering the relative positions of patches in an image [8]. When working with video and/or audio, we can cite visual-audio correspondence verification [21], where one of the two modalities is used as a supervision signal for the other modality and temporal context structure [29], where the supervision signal is the temporal order of the frames in a video.

In document information retrieval, the authors of [4] propose an interesting and promising approach using clustered SIFT descriptors ([25]) as *pretext-task* labels for writer retrieval. The authors of [33] generate simulated-shredded documents to train a model for reconstructing mixed strip-shredded text documents. An other interesting approach is proposed in [9], where the authors apply various random transformations to the unlabelled source images to create a variety of images that compose a surrogate class, then used for supervised training.

3 Datasets

In the field of ancient documents reconstruction, it is very difficult to find databases of scales sufficient to train deep neural networks. This is why we chose to contribute by building our own learning database based on the Papyrus collection of the University of Michigan. A significant pre-processing phase is necessary in order to extract a useful working database for deep learning. This pre-processing is detailed in section 3.2.

Another useful source of data comes from the 2020 ICFHR HisFragIR20 competition [40] [2]. The competition provides a training set containing about 17222 labeled document from scans of ancient documents that have been artificially torn into 100000 fragments. The test set is composed of 2732 documents artificially torn into 20019 fragments

[2] https://lme.tf.fau.de/competitions/hisfragir20-icfhr-2020-competition-on-image-retrieval-for-historical-handwritten-fragments

The two databases are very different quantitatively and qualitatively. The Hisfrag dataset is much larger and the numbers of fragments per papyri is different, as can be seen in Table 1 and Fig. 4. Moreover, there are differences in the aspects of the fragments themselves between the Michigan dataset, the Hisfrag train dataset and the Hisfrag test dataset, as can be seen on Figs. 5 and 6.

	Hisfrag train	Hisfrag test	Michigan
Nb papyri	17222	2732	1118
Nb fragments	101706	20019	4579
mean frags/papy	5.9	7.3	4.4
std frags/papy	9.7	2.9	6.4

Table 1 Summary of the number of papyri and fragment for each datasets (after pre-processing)

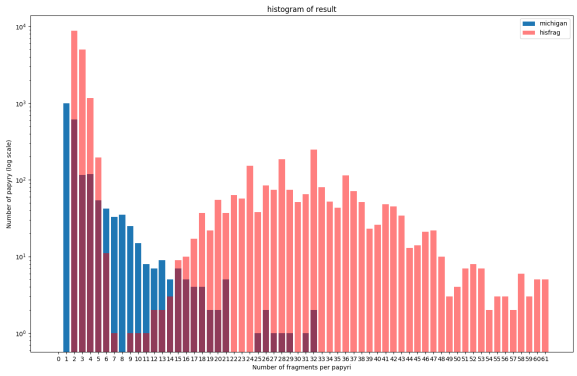

Fig. 4 Distribution of the number of fragments per papyri on the Michigan and Hisfrag databases. In red, the Hisfrag database and in blue the Michigan database

3.1 The Hisfrag database

The Hisfrag database does not need any pre-processing, it is given by the authors as a collection of images whose file names encode the labels.

The database was created by building artificial fragments from document images using an algorithm explained in [40]. As can be seen on Fig. 6, the training and test sets look very different. This is because the authors chose to make them from different images sources. The images from which the training set is made come from the Historical-IR19 [5] dataset, while the images from which the test set is made are from the University

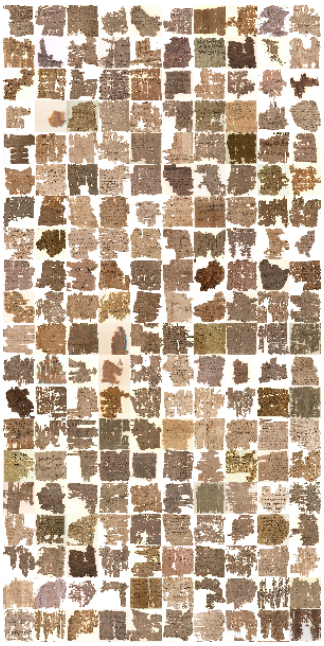

Fig. 5 Random sampling of fragments in the Michigan dataset

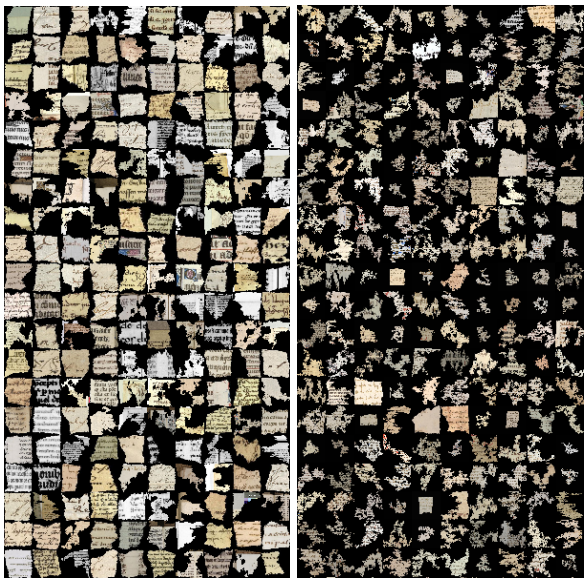

Fig. 6 Random sampling of fragments in the Hisfrag train dataset (left) and the Hisfrag test dataset (right)

Library Basel [3]. See Table 1 for numerical details on both datasets.

This makes for a very challenging dataset, which is reflected by the results of the competition. Indeed, the winner of the competition only achieved 22.6% of mean average precision, a 36.4% Top-1 Accuracy, a $Pr@10$ of 31.2% and a $Pr@100$ of 58.9% on the page retrieval task.

[3] https://www.unibas.ch/

3.2 Pre-processing the Michigan database

The Michigan papyrus collection is public domain and free to use for educational and research purposes under a *Creative Commons Attribution 4.0 license*. It contains a total of 17029 images of documents written in different languages. We choose to only use the 14890 Greek Papyri to avoid the possibility of a language related bias when learning and because it is by far the most represented language in the database, the second most represented being Coptic with only 1140 images.

From these 14890 images, we remove 515 negative and infrared images. We also remove 8185 duplicate images that are different images of sub parts of the same papyrus. At this step, we are left with 6190 color images.

The fragments can either be already separated in different images, or be in the same image. In the latter case, we have to separate them in different images. The images we get from the collection each contain a ruler and a color reference scale that we also have to remove. See Fig. 7 and Fig. 8 for an example of pre-processing.

We also want to know the real size of the papyri (in centimeters) in order to rescale them all at the same resolution (pixel density). As the images of the database have been taken at different times, with different cameras and different zoom levels depending on the size of the considered papyrus, we use the ruler and the color scale (See Fig. 7). Making the assumption that their physical size is the same across every pictures, we determine that the height (or width, depending on the orientation) of one color rectangle of the scale is 23mm by looking at the ruler next to it. Then, we use a simple color matching heuristic to isolate a color rectangle in each image, compute the contours of the shape and get a geometric representation[4]. At this point, we can determine the orientation and length of the rectangle in pixels, and simply divide this length by its real length in centimeters to obtain the pixels per centimeters density value. We store the pixels per centimeters value of each image in a json file.
After separating the fragments contained in single images, we are left with 6607 images of fragments, which belong to 3823 papyri. Within the remaining papyri, we are only interested in the ones that are composed of at least two fragments. As can be seen on Fig. 4, papyri composed of only one fragment are a significant portion of the dataset, but cannot be used as real ground truth data. As in the Hisfrag dataset, we do not include such papyri in our dataset. In the end, we are left with 1118

[4] using OpenCV : https://opencv.org/, accessed November 3, 2020

papyri composed of 4579 fragments, each papyrus being composed of at least two fragments.

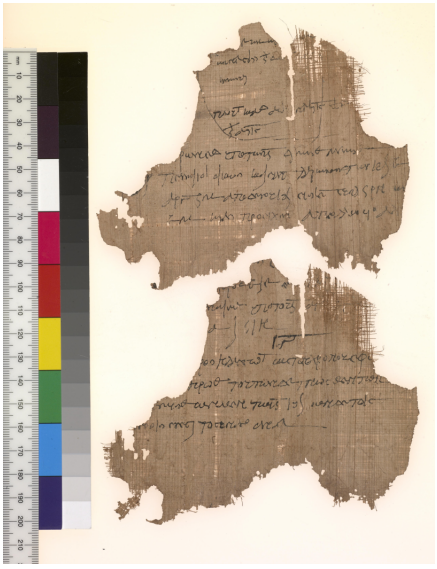

Fig. 7 Pre-processing the Michigan database : The original image

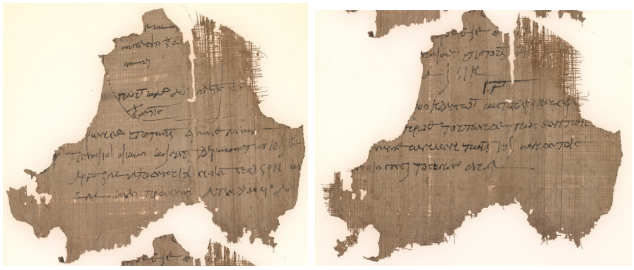

Fig. 8 Pre-processing the Michigan database : The fragments extracted into two images

4 Learning a similarity score using Deep Metric Learning

4.1 Global architecture

In order to learn a similarity score between two inputs, we use Siamese Neural Networks. After the feature extraction parts, we chose to compute the absolute element-wise difference of the two output vectors - the embeddings - and to feed this difference into two dense layers of size 512 that are connected to a sigmoid neuron as the final output. We also tried to concatenate the output vectors, but noticed it produced a symmetry issue : swapping the input images resulted in different final output scores. Using dense layers with a sigmoid output allows us to optimize using the *Binary Cross Entropy* loss function. It would also have been possible to use a loss function that optimizes directly the distance between the two embeddings, such as the *Contrastive Loss* [11]. As the two approaches coexist in the literature, we decided to use the first one. The output corresponds to the probability of the two inputs being similar (1 : similar, 0 : dissimilar). The loss function then uses this output to guide the optimizer in tuning the weights of the network with the goal of minimizing the score when the two inputs are dissimilar, and maximizing it when they are similar. Fig. 9 illustrates the architecture.

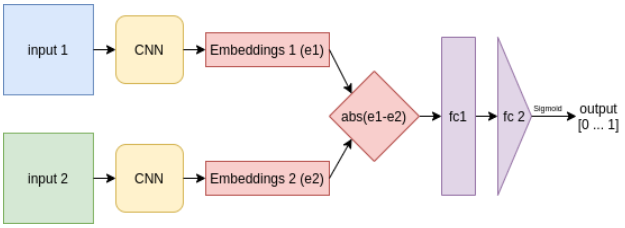

Fig. 9 An abstract illustration of our siamese models

As said in Section. 2, triplet networks and/or triplet type losses may improve the performance of a model compared to pure siamese approaches. They have in particular shown their effectiveness in the people re-identification task [39]. As the main contribution of this paper is a method that can be used to work with little to no labeled data, we decided to focus our testing on the capacity of our approach to learn in a self-supervised way with or without fine-tuning, this is why triplet networks were not considered and we decided to test two convolutional networks (VGG16 and ResNet50) and one architecture : the Siamese Network.

4.2 Testing different architectures

When working with images, the branches of the Siamese Network most often are convolutional neural networks with flattened outputs. Thus, we can use any state of the art convolutional neural network by removing the final dense layers and keeping only the convolutional parts (the feature extraction part). In Fig. 9, this means replacing the two *CNN* blocs by the same convolutional part of different networks. In this work, we use the convolutional parts of *VGG16* [23] and *Resnet50* [12]. After flattening, the dimensions of the 1D layers are respectively 2048 and 8192. We chose these two architectures for their tried and tested feature extraction abilities, and their convenient availability in the Deep

Learning framework we work with.

VGG16 Architecture
VGG16 is a conventional (to today's standards) convolutional neural network stacking convolutional and max pooling layers with increasing feature maps depth and fully connected layers at the end. It was designed for the *ImageNet* dataset [7] and obtained state of the art classification accuracy at the time of its publication. It is still a widely used architecture today, as it is simple, available in many deep learning frameworks and yields good results.

Resnet50 Architecture
Deep Residual Networks such as *Resnet50* were introduced as a solution to the vanishing gradient problem that occurs with neural networks getting deeper, limiting the learning abilities of such deep networks. They work by introducing *Residual Blocks* to the network architecture, which consists in adding a "shortcut" connection between layers. The authors [12] demonstrated that much deeper networks could be successfully trained using this method than without.

4.3 Batch constitution

The composition of the mini-batches and the patch extraction process are critical aspects of our training methodology. We discuss here why we have to build balanced mini-batches for training as well as why and how we select the most relevant patches to represent the fragments.

Balancing the mini-batches

In order to train the Siamese Networks, we construct similar input pairs and dissimilar input pairs. During training, the pairs are fed to the network, a similarity score is outputted which we compare to the ground truth, 1 if the pair is similar and 0 if the pair is dissimilar, to compute the loss function.

Here, an issue arises : if we build all the possible pairs, there are far more dissimilar pairs than similar pairs. This is due to the nature of the data we work with, there are many classes and few samples in each classes.

If the network is given an immense majority of dissimilar pairs during training, it will only learn to always output the same result [15]. To tackle this issue, we chose to build balanced mini-batches during training, i.e. each mini-batch contains as many similar pairs as dissimilar pairs. This is effectively equivalent to either over-sampling the minority class or under-sampling the majority class.

Another solution is to weight the loss function, giving more importance in the computation of the loss to the minority class. However, we noticed empirically that weighing the loss function performed worse than the balanced mini-batches solution on our data and models.

Extracting the most relevant patches

We use a patch based approach to train our models. The idea is to represent a fragment as a collection of patches which are, hopefully, representative of the full fragment. This also allows us to control the patch extraction process to remove as much background and borders as possible as we can't rely on the shape of the fragments (overlapping fibers, degraded contours, missing fragments ...). Moreover, this makes for a simpler way to feed the data to the networks, as all the inputs are of the same size, while the full images vary a lot in size and shapes. As well as these data specific and practical considerations, Bondi et al. [1] suggests that this approach should yield very similar performance as using full size images (as long as a sufficient number of well chosen patches are selected), while greatly reducing the number of features extracted per training sample making the task of the classifier easier.

We describe here our process for extracting useful patches from the fragment images. First, we extract as many non-overlapping patches of size 64x64 pixels as possible for each image. There is no overlap between fragments in the data as they are torn papyri. In order to be representative of the actual data, the patches need to be non overlapping. This also forces the network not to learn some kind of local pattern matching. Then, we rank the patches based on their content. We make the hypothesis that patches containing more text are more informative than patches containing more papyrus texture. Consequently, we designed a score function based on the proportion of background (i.e. the part of the image that is not papyrus) and the proportion of text. The score function is defined as follows :

$$patchScore = \frac{nbTextPx}{totalNbPx} + (1 - \frac{nbBackgroundPx}{totalNbPx})$$

Where $nbTextPx$ is the number of pixels that constitute text, $nbBackgroundPx$ is the number of pixels that constitute background and $totalNbPx$ is the total number of pixels in the patch.

Finally, we select the n patches with the highest scores. In the end, we have n patches representing each fragment. We empirically chose $n = 5$, as it seems to

be representative enough to obtain decent results, while not being too large. Indeed, when evaluating the models, we have to compute the similarity score on all possible pairs of patches. Choosing a higher n value results in a quadratic increase of the number of computations, which quickly results in an enormous amount of computation time. Fig. 10 illustrates the patch extraction and labeling process with *papyrus-level* information available (we know from which papyrus a fragment comes from).

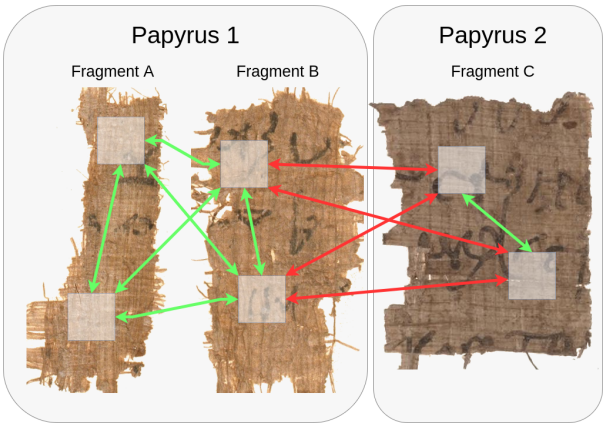

Fig. 10 Patches extraction and labeling process with *papyrus-level* information. Green and red links indicate respectively a similar pair and a dissimilar pair.

4.4 Evaluation method

We evaluate our models on a hold-out test set that has never been used during training. As explained previously, during training we have to balance the batches for the network to converge. However, when evaluating the performances of the models for a retrieval task, we have to consider all the possible fragments pairs in the test dataset. For each fragment, we compute the score between this fragment and every other fragments. This score is computed by taking the average of the scores of each pair of patches between these two fragments. With this, we build a 2D "similarity matrix" for all the fragments of the test set and a similar "ground truth matrix" telling for each pair of fragments if they belong or not to the same papyrus. From these matrices, we can compute all the metrics described below.

4.4.1 Hisfrag competition metrics

The Hisfrag competition used 4 metrics to evaluate the performances of the competitors in the retrieval task.

Mean Average Precision (mAP) : Each fragment is compared with every other fragment : this is a query. The average precision of a query is defined as follows :

$$AP = \frac{1}{R}\sum_{r=1}^{S} Pr(r) \times rel(r)$$

Where R is the number of relevant fragments, S is the size of the query (list of scores), $Pr(r)$ is the precision at rank r and $rel(r)$ is a function returning 1 if the fragment at rank r is relevant and 0 otherwise. We take the average of the APs returned for each query to obtain the Mean Average Precision (mAP). This metric gives an idea of how good the model is at ranking similar examples higher than dissimilar examples. A high mAP means a greater chance of the best ranked fragments begin relevant fragments.

Top-1 Accuracy : We look at the best ranked fragment for each query. If this fragment is relevant, the score for this query is 1, otherwise it is 0. We take the average of this score for each query to get the Top-1 Accuracy. This metric gives an indication of what is the probability that the best ranked fragment is a relevant fragment, which is very useful for papyrologists.

Precision at 10 and 100 (pr@10 and pr@100) : For each (sorted) query, we compute :

$$Pr@k = \frac{R_k}{min(R_N, k)}$$

Where R_k is the number of relevant fragments up to rank k and N is the total number of fragments in the query. We average the scores of each query to get the global $Pr@k$ score. The meaning of the $min()$ denominator is that if we retrieve all the relevant fragments, we want a perfect score (a score of 1), so we divide by the minimum between the total number of relevant fragments R_N and the size of the query k. This metric gives an indication of the proportion of relevant fragments retrieved for a query of size k.

4.4.2 Metrics relevance for evaluating the task

Choosing the right metrics for the task we want to evaluate is critical in order to get a good idea of the performances of the models, particularly when dealing with a large imbalance between classes. We are in this case, as there is a much greater number of possible dissimilar pairs than of possible similar pairs. The metrics chosen for the Hisfrag competition are relevant and standard for the retrieval task we are tackling [27]. One can note that the Average Precision (AP) of a query approximates the area under curve of the precision/recall curve

[40], which is a good metric in our context according to [38]. Precision at k is also relevant, especially from a usability by experts perspective as it gives an indication of the expected number of relevant associations among the top k documents.

Conversely, using a *ROC* plot in this context can be misleading [38]. Similarly, looking only at the accuracy for a given decision threshold could be extremely misleading. If we imagine a model that has only learned to always output that the two images are dissimilar, having (for example) 1000 dissimilar pairs and 100 similar pairs in the test set would still yield an accuracy score over 90%.

5 Baseline, with and without domain adaptation

5.1 Baseline results

For computation time reasons, all the results described in this section were evaluated on subsets of the test datasets. For each dataset, we use 100 *papyri, which constitute a set of about* 800 *fragments for the Hisfrag dataset, and a set of about* 450 *fragments for the Michigan dataset.*

As a baseline, we train and evaluate our models on the two datasets in a "normal" way, meaning without *fine-tuning* nor *self-supervised learning*. For the Hisfrag dataset, we split 10000 papyri ($\approx$ 60000 fragments) into 9000 training papyri ($\approx$ 54000 fragments), 1000 validation papyri ($\approx$ 6000 fragments) and we evaluate on the separate Hisfrag test set. For the Michigan dataset, we split 1000 papyri ($\approx$ 4500 fragments) into 800 training papyri ($\approx$ 3500 fragments), 100 validation papyri ($\approx$ 450 fragments) and 100 test papyri ($\approx$ 450 fragments) for evaluation. On both datasets, we extract 5 patches of size 64×64 per fragment. Patches of size 32 are too small to contain full characters, and patches of size 128 don't allow us to fit enough images in the GPU memory for efficient training.

We train for 100 epochs for each model with balanced mini-batches of size 128. The optimizer is Adam [19] with a decreasing learning rate (starting at 0.001, ending at 0.00005). In addition to looking at training and validation accuracies during training, we compute a histogram of the scores returned by the model at each epoch on a validation batch (see Fig. 11). This allows a quick visual understanding of how good the model is at separating similar pairs from dissimilar pairs. In the example on Fig. 11, the model learned to correctly attribute high scores to similar pairs, and low scores to dissimilar pairs most of the time.

		mAP	top-1	pr@10	pr@100
His	VGG16	**0.67**	**0.87**	**0.75**	**0.92**
	Resnet50	0.61	0.83	0.71	0.82
Mich	VGG16	0.41	**0.68**	0.47	0.85
	Resnet50	**0.54**	0.67	**0.59**	**0.92**

Table 2 Baseline results. Here, we directly train for the task of determining if two fragments come from the same papyrus, therefore using the *papyrus-level* information available. The results were computed on about 800 fragments from the Hisfrag dataset, and about 450 fragments from the Michigan dataset.

Looking at Table 2, we see that the models perform differently depending on the datasets. *VGG16* seems to work best with the Hisfrag dataset while *Resnet50* seems to work better on the Michigan dataset. Globally, the results are quite worse when working with the Michigan dataset than with the Hisfrag dataset, even though the data seems more challenging because of the differences between the global aspect of the fragments in the train and test sets. This is most likely an effect of the magnitude of data used for training the models, obviously more data is better.

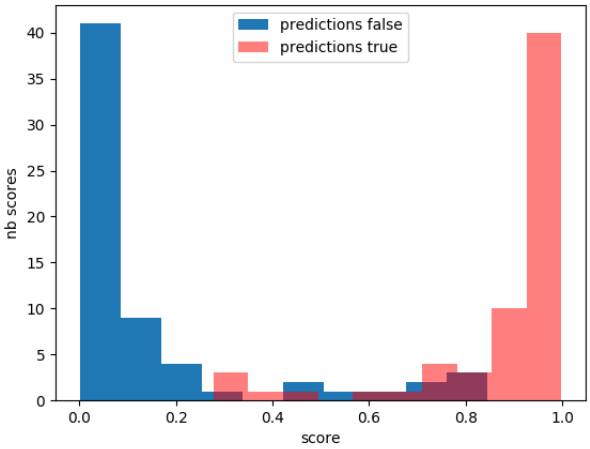

Fig. 11 Example of histogram of the scores computed on a batch (size 128) of the validation set during training of the *VGG16* model on the Hisfrag dataset. There are actually two histograms. The red and blue histograms correspond to scores that were attributed to sample pairs that where respectively similar and dissimilar

The *VGG16* based model trained and evaluated on the Hisfrag dataset reaches an average top-1 score of 0.87. This means that on average, for a given query, the probability of the best ranked fragment being relevant is 87%. To put this into perspective, the expected probability of the best ranked fragment being relevant when ranking randomly is $\frac{7.3}{800} = 0.009125 \approx 0.9\%$. With 7.3

being the average number of fragments per papyrus on the *Hisfrag* test set, and 800 being the number of fragments considered in this test.

The relatively low *mAP* scores indicate that there is a non-negligible amount of non-relevant fragments that are ranked better than relevant-fragments. A high false positive rate is expected by the sheer number of comparisons that are being computed. Indeed, with 800 fragments, we perform $800 \times 799 = 639200$ predictions. Even assuming a theoretical *quasi-perfect* model that would mistakenly give a high relevance score to a non-relevant fragments pair in 0.1% of the cases, we would still have $639200 \times 0.001 = 639.2$ false positives. This is why we are more interested in the *top-1* and *pr@k* scores, as they are a better reflection of the real-world usability of the models by papyrologists trying to reconstruct the papyri.

In order to give a correct interpretation of the *pr@10* and *pr@100* scores, we have to remember that, for example, the Hisfrag test subset we compute the scores on is of size 800. Thus, a single query is made of 799 comparisons (we obviously don't compare the fragment to itself). On average, we expect 7.3 comparisons (see Table 1) to be *positive* (i.e. the compared fragment is relevant).

Given the definition of the *pr@k* metric explained in section. 4.4.1, the *pr@k* score could have two meanings. On the one hand, if k is inferior to the actual number of relevant fragments to be retrieved, we are computing the proportion of relevant fragments retrieved with respect to k. In this case, a score of 1 means the first k fragments retrieved are all relevant, but other relevant fragments could exist on lower ranks. On the other hand, if k is superior to the actual number of relevant fragments to be retrieved, we are computing the proportion of relevant fragments retrieved with respect to the actual number of relevant fragments to be retrieved. In this case, a score of 1 means that all the possible relevant fragments have been retrieved within the first k fragments, but non-relevant fragments could exist within these k fragments.

Going back to the results presented on Table 2, a *pr*@10 score of 0.75 on the Hisfrag dataset with the *VGG16* based model means that within the 10 best ranked fragments, we have recovered on average 75% of all the relevant fragments (assuming 7.3 fragments per papyri on average). We can give the same interpretation for the *pr*@100 score of 0.92, within the 100 best ranked fragments, we have recovered on average 92% of all the relevant fragments.

Fig. 12 shows four samples of queries results on the Hisfrag test database. We can see that the model seems to work better on fragments that are less degraded (columns 1 and 3). This is not surprising, as we see on Fig. 6 that the Hisfrag training set is mainly composed of less degraded fragments (first column), while the Hisfrag test set contains a majority of more degraded fragments.

Fig. 12 Examples of 4 queries up to rank 10. For each query (in columns), the first fragment (outlined in blue) is the query fragment and the others are the 10 best ranked fragments by decreasing score. Outlined in green and red are respectively relevant and irrelevant fragments. The images of the fragments have been resized without keeping their aspect ratio for this visualization.

5.2 Using a model trained on another dataset

A common approach when working with a small dataset is to use the "learning power" of a larger dataset to pre-train a model, then *fine-tune* this model with data from the smaller dataset. The goal is to learn robust low level feature extractors (the first convolutional layers) on the large dataset, and retrain the classification part (the fully connected layers) using some of the data we are interested in. In general, this is called *transfer-learning*, or knowledge transfer. For example, it is very common to "kickstart" the learning process ([35], [45], [14]) by using a model pre-trained on *ImageNet* [7], taking advantage of the robust convolution kernels learned from such a large dataset. The authors of [41] report improved performance when using transfer learning from *ImageNet* in the context of content-based image retrieval. After some experiments, we could not get the models to converge as well using pre-trained with *ImageNet* weights as when training from scratch. We decided to leave it aside for now, as this is an optimization whose absence does not impact the validity of the approach. We are more interested in the sub domain of *transfer-learning* that is *domain-adaptation*. This is the same idea, but using data from the same *domain* to create the pre-trained model to fine tune with our data. *Domain-adaptation* techniques can be unsupervised, semi-supervised of fully supervised depending on the availability and amount of target data. Different approaches exist, including iteratively labeling the target data with "pseudo-labels" [42] and using *adversarial learning* where the the features from both source and target dataset are pushed into a common latent space where they are optimized to be indistinguishable [3].

We first evaluate the ability of each model pre-trained on the Hisfrag dataset to predict on the Michigan dataset as is. We take the models trained in section 5.1 on the Hisfrag dataset and apply our evaluation protocol on the Michigan dataset in the same conditions. We see on the first two lines of Table 3 that this simple approach yields much worse results than when predicting using a model trained with the Michigan dataset as shown in the baseline results in Table. 2.

Then, we select a small number of annotated data (50 papyri, $\approx$ 220 fragments) from the Michigan dataset to fine tune the pre-trained models (*domain-adaptation*). We chose to use a small number of papyri because in order to be useful to papyrologists, the methods we propose have to require as little human labeling work as possible. This approach would not be very useful in a real case scenario if the experts had to label thousands of their images for the method to work. For the sake of completeness however, we also applied the fine-tuning process with 1000 Michigan papyri. We initialize the networks with the weights from the pre-trained models, freeze the convolutional layers and start training with 50 and 1000 papyri. We then evaluate in the same conditions as before.

Here, the results improved compared to above, but are still quite far from the baseline. Interestingly, and contrary to what could have been expected, performing fine-tuning with as many papyri as in the baseline did not improve the results here, but rather degraded them quite significantly for both architectures (third line of Table 3). Indeed, with a maximum *mAP* of 0.45 with *Resnet50*, there is almost a 10 points reduction of performance compared to the baseline *mAP* of 0.54%. We can also note that going from 50 to 1000 papyri led to a marginal improvement with *VGG16*, but a significant one with *Resnet50*. Further study would be needed to understand such different behaviours from the two architectures. However, as this approach requires access to a large annotated dataset for pre-training and non negligible expert annotation work to generate the data for fine-tuning, we chose to explore another approach. In the next section, we show that we can get even closer to the baseline by training the models in a *self-supervised* way, removing these two constraining requirements.

		mAP	top-1	pr@10	pr@100
no fine tuning	VGG16	0.21	0.37	0.29	0.59
	Resnet50	**0.26**	**0.47**	**0.31**	**0.63**
fine tuning	VGG16	**0.34**	**0.60**	**0.41**	**0.79**
	Resnet50	0.31	0.56	0.37	0.72
fine tuning 2	VGG16	0.35	0.62	0.42	0.81
	Resnet50	**0.45**	**0.74**	**0.49**	**0.86**

Table 3 Evaluating the models trained on the Hisfrag dataset on the Michigan dataset without and with fine-tuning. First line without fine tuning, second line is fine-tuned with 50 papyri and last line is fine-tuned with 1000 papyri. (fine tuning 2)

6 Self-supervised learning

In this section, we use the terms *papyrus-level* information and *fragment-level* information. Having *papyrus-level* information available indicates that we know the "real" ground truth, meaning that we know from which papyrus comes a given fragment (and consequently, we know from which papyrus comes a given patch). Having only *fragment-level* information means that we only

know from which fragment comes a patch.

The baseline results presented before assume that ground truth at the *papyrus-level* is available for training. In a real-world situation however, papyrologists working on new-found papyrus fragments need solutions that requires as little manual annotation as possible for these solutions to be useful. We explore the possibility of training the models in a context where no *papyrus-level* ground truth is available. We use a *self-supervised learning* approach. In this domain, we call the *pretext task* the task we are solving during training, which is different from the *target task* that is the task we ultimately want to solve. Solving the *pretext task* needs to require an understanding of the data that will be relevant to solve the *target task* [20]. The power of this approach comes from the possibility of designing a *pretext task* from the data itself, without additional labeling. Here, our *pretext task* is to determine if two patches come from the same fragment, and our *target task* is to determine if two fragments come from the same papyrus. To build the *pretext task* dataset, we label the patches with the identifiers of the individual fragments, rather than from the papyri themselves because this information is unknown in this case (see Fig. 13). This has the side effect of mislabeling a small portion of the possible pairs with respect to the actual ground truth (dotted red lines on Fig. 13).

We first estimate the amount of mislabelings induced by the *self-supervised* ground truth generation approach. We then experiment with different versions of this approach, with an emphasis on being true to the real conditions in which the papyrologists would use the models.

6.1 Estimating the quantity of mislabelings

Looking at Fig. 13 we see that patches 1 and 2 belong to the same fragment, so they are considered "similar" (green link) while patches 1 and 3 belong to different fragments so they are considered dissimilar (dotted red link) even though the two fragments come from the same papyrus. Patches 4 and 5 are rightly considered dissimilar because they come from different fragments that belong to different papyri.

We build our balanced training batches by first sampling a random patch in the whole pool of patches. Then, we sample a similar patch (a patch coming from the same fragment) half of the time, and a dissimilar patch half of the time. When sampling the dissimilar patch, it is possible to sample a patch coming from the same papyrus, but a different fragment, this would be

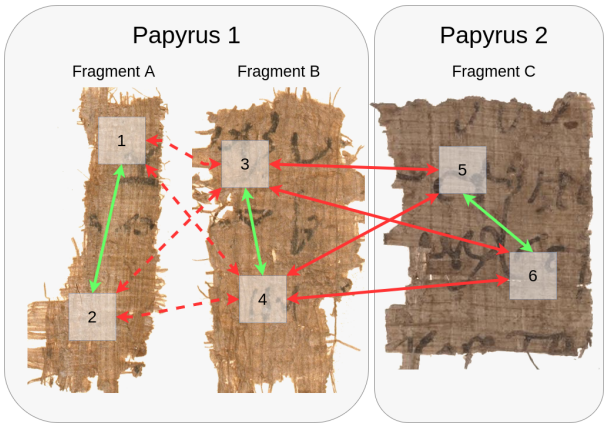

Fig. 13 Patches extraction and labeling process in "self-supervised learning mode". Green and red links indicate respectively a similar pair and a dissimilar pair. The dotted red lines indicate that these links actually are labeling errors with respect to the ground truth.

a case of a mislabeled pair (e.g. patches 1 and 3 in Fig. 13). We can compute an estimation of how many pairs will be statistically mislabeled this way:

Let's take 10000 papyri in the Hisfrag train dataset. According to Table 1, we will get on average $10000 \times 5.9 = 59000$ fragments because each papyrus contains 5.9 fragments on average. If we extract 5 patches per fragment, we have on average $5 \times 5.9 = 29.5$ patches per papyrus, and in total, we have $5 \times 59000 = 295000$ patches. The probability of sampling a patch coming from the same papyrus but not from the same fragment (mislabeling) is the following :

$$P_{mis} = \frac{29.5 - 5}{295000 - 5} = 8.3e - 05$$

With a batch of size 128, we sample 64 dissimilar pairs, so when creating a batch, the probability of it containing a mislabeled pair is :

$$P_{mis-batch} = P_{mis} \times 64 \approx 0.005$$

As this probability is very low, we hypothesize that it should not affect the training performances much.

6.2 Self-supervised learning vs domain adaptation

First, we train each model from scratch with *fragment-level* ground truth, however we evaluate at the *papyrus-level*, as before. We evaluate in the same way as with previous experiments from section 5.1, only, the models never had access to *papyrus-level* ground truth during training. With this experiment, we evaluate the ability of the models to generalize the *target-task* from the *pretext-task*. That is, given only *fragment-level* information, can the models learn to determine if two fragments

belong with each other. Table 4 shows the results with the models trained on the same amount of data as section 5.1 Table. 2.

		mAP	top-1	pr@10	pr@100
His	VGG16	0.38	**0.73**	0.47	0.72
	Resnet50	**0.41**	0.51	**0.48**	**0.78**
Mich	VGG16	**0.38**	**0.61**	**0.43**	**0.84**
	Resnet50	0.30	0.43	0.39	0.76
Mich 2	VGG16	0.32	0.32	0.38	0.78
	Resnet50	**0.43**	**0.75**	**0.47**	**0.84**

Table 4 *Self-supervised* learning results. The models were trained with *fragment-level* information only, and evaluated on the *papyrus-level target-task* with 100 papyri. First line on the Hisfrag dataset, second line on the Michigan dataset. The third line corresponds to the Hisfrag model fine-tuned using only *fragment-level* information from the Michigan dataset.

It is not surprising to obtain worse results in *self-supervised* mode than when training with the *papyrus-level* information as in section 5.1 (on average, -0.19 for the *mAP*, -0.19 for the *top-1*, -0.19 for the *pr@10* and -0.05 for the *pr@100*) as the latter directly optimizes for the *target-task*. However, it is interesting to note that the *self-supervised* approach performs a little bit better than the *domain adaptation* approach fine tuned with 50 papyri ($\approx$ 220 fragments) with *papyrus-level* information reported in Table. 3. Indeed, the approach yielding the results of Table. 3 required a pre-training step with a large dataset, then a fine-tuning step on 50 annotated papyri, while the *self-supervised* approach is standalone (as it does not require pre-training on another annotated dataset) and does not require any annotation on the data we want to predict on. We see again that the Hisfrag dataset seems to perform better due to difference in the amount of training data, but interestingly the gap in performance with the Michigan dataset is smaller. This could be an effect of the differences in aspect between the training and testing data of the Hisfrag dataset which is exacerbated here.

We can notice another interesting thing when comparing the third line of Table. 4 with the third line of Table. 3. On Table. 4, the same pre-trained model is used as in Table. 3, but the fine-tuning is done using only *fragment-level* information. The performances obtained are very close (for *Resnet50* for example, -0.02 for the *mAP*, -0.01 for the *top-1 accuracy*, -0.02 for the *pr@10* and -0.02 for the *pr@100*), meaning that in this case, it is a lot more interesting to use the self-supervised approach combined with the pre-trained model on the Hisfrag dataset rather than having to annotate 1000 papyri.

Fig. 14 shows sample queries on the Hisfrag test dataset using the *Resnet50* model trained on Hisfrag (with 0.41 *mAP*). We see on these samples that the query images are challenging in the sense that they are quite degraded, but the *self-supervised* model still manages to rank in the top-10 at least one matching fragment (framed in green) from the 799 fragments in

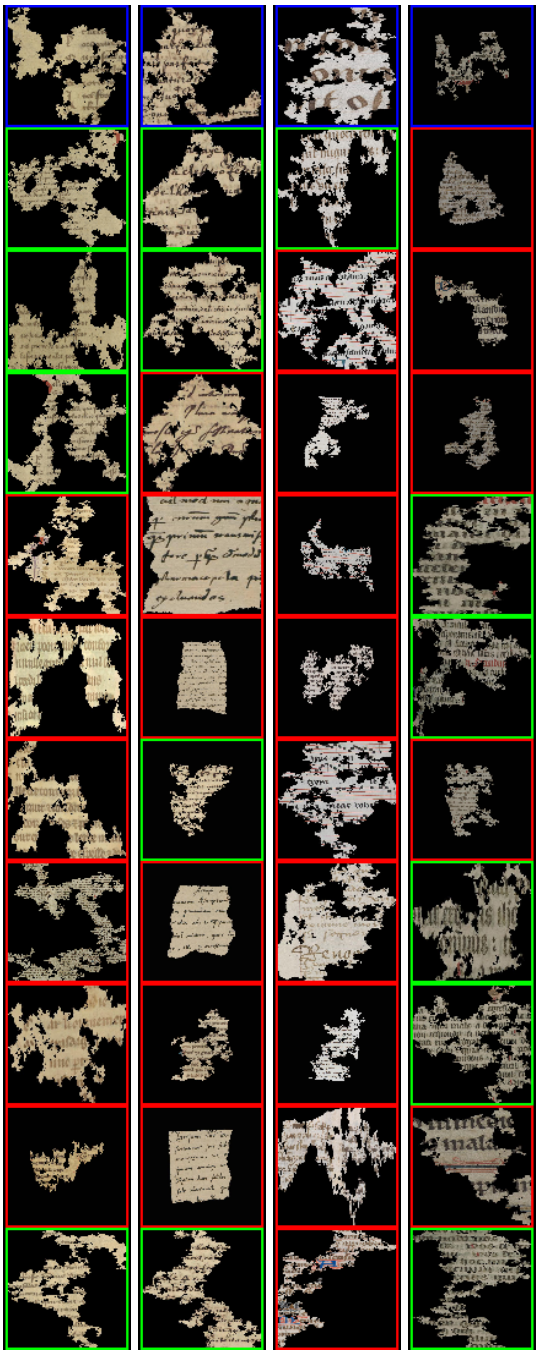

Fig. 14 Examples of 4 queries up to rank 10. For each query (in columns), the first fragment (outlined in blue) is the query fragment and the others are the 10 best ranked fragments by decreasing score. Outlined in green and red are respectively relevant and irrelevant fragments. The images of the fragments have been resized without keeping their aspect ratio for this visualization.

the query. Moreover, even the false positives (framed in red) actually look quite similar to the query fragment (at least to a moderately trained eye), which testifies to the challengingness of the data.

6.3 Exploring a more realistic use case

The most useful approach for papyrologists would be to be able to learn a model in a *self-supervised* fashion on the unlabeled data they want to work on, and to extract predictions on this very data. Contrary to the previous experiments, we experiment here with training from scratch with different fragments corpus sizes, and predicting on the same data the models were trained on. Here we are not trying to estimate how well the model has generalized on the *target-task*, but rather how it can infer *papyrus-level* matches with limited *fragment-level* only information as training examples.

We have to keep in mind two effects working against each other. The more data we use for training, the better the models should perform, but also the harder the retrieval task is. According to the results reported in Table. 5, on the Michigan dataset, there seems to be a "sweet spot" for the *VGG16* model around 400 fragments where it had enough data to make sensible predictions, and a sufficiently low number of retrieval candidates. When increasing the number of fragments, we notice a drop in retrieval performance. This is an effect of the lack of training data relative to the scale of the retrieval task, we can compare the last two lines of Table. 5 with the last two lines of Table. 4 to corroborate this, as in both cases we evaluate on 100 papyri. In the latter, the models were trained using 800 papyri which amounts to about 3500 fragments, which is almost 5 times more data as in the former.

With an average *top-1* accuracy score of 0.70 with *VGG16* when using about 400 fragments, this configuration seems to be the most promising for papyrologists who want to use the *self-supervised* approach. However, it seems that this approach does not scale well with the increase of available data, as the additional "learning power" it provides does not compensate for the increased retrieval difficulty it provokes.

For the sake of completeness, we do the same experiment (Table 6) with data from the Hisfrag train dataset to see how the approach performs on a dataset with different properties (different kind of documents, different number of fragments per papyri, different fragment sizes etc). We see that with 25 papyri, the approach yields good results, with a *top-1* accuracy of 0.75 and a *pr@100* of 0.96. Such a high *pr@100* score means that almost all fragments that had to be retrieved within the 542 fragments of the query have been ranked in the top 100, which is a very interesting property. However, it is difficult to compare the two experiments, as the number of fragments per papyri is higher in the Hisfrag dataset than in the Michigan dataset. This first means that for the same number of papyri, there is more training data available on the Hisfrag dataset, which improves the robustness of the trained model. This also means that for the same number of fragments, there is a lower probability of mis-association on the Hisfrag dataset than on the Michigan dataset. But globally, with more fragments more comparisons are computed, which increases the likelihood of false positives due to the non-perfect accuracy of the models. Given the results we could obtain on these two datasets which contain different kinds of documents, different numbers of fragments per document and different average fragment sizes, we feel confident in saying that the approach should transpose well to other datasets. Further study would be needed to determine what factors in the data influence the most the quality of the predictions in order to provide more precise guidelines depending on data specifics.

	paps	frags	map	top-1	pr@10	pr@100
vgg	25	213	0.38	0.61	0.43	**0.90**
res	25	213	0.33	0.41	0.44	0.83
vgg	50	394	**0.44**	**0.70**	**0.52**	0.87
res	50	394	0.27	0.40	0.36	0.71
vgg	75	651	0.23	0.4	0.29	0.63
res	75	651	0.23	0.27	0.31	0.67
vgg	100	735	0.12	0.21	0.15	0.44
res	100	735	0.19	0.25	0.27	0.56

Table 5 Realistic use case on the Michigan dataset. The models are trained in *self-supervised* mode and the scores are computed on the same data, but with *papyrus-level* ground truth.

	paps	frags	map	top-1	pr@10	pr@100
vgg	25	542	**0.63**	**0.75**	**0.66**	**0.96**
res	25	542	0.57	0.64	0.61	0.91
vgg	50	832	0.39	0.59	0.42	0.8
res	50	832	0.5	0.57	0.58	0.86
vgg	75	1479	0.31	0.54	0.34	0.66
res	75	1479	0.44	0.56	0.52	0.78
vgg	100	1912	0.37	0.64	0.44	0.7
res	100	1912	0.51	0.62	0.6	0.81

Table 6 Realistic use case on the Hisfrag dataset. The models are trained in *self-supervised* mode and the scores are computed on the same data, but with *papyrus-level* ground truth.

We argue that this approach can be used as a matching suggestion tool with expert supervision, as even the worst results we obtained still significantly reduces the manual search space for the expert, with 15% of the relevant fragments in the *top-10* and 44% of the relevant fragments in the *top-100* (second to last line in Table. 5), avoiding the need for manually analyzing 735 fragments in order to retrieve some relevant fragments in this case.

On this topic, our research is part of the *GESHAEM* project, a five years scientific research initiative funded by the European Research Council (ERC) aiming at studying the *Jouguet collection* of the Sorbonne [5] for a better understanding of the early Ptolemaic administration. The process of digitizing and reconstructing the papyri of this collection is still in progress, so we would like to provide the papyrologists working on it with matching suggestions using the methods we developed. The quantity of data that will be available after the digitization campaign should be of the same order of magnitude (less than 1000 fragments) as the experiments that worked best here. It will be interesting to see if our predictions are relevant to the papyrologists and if they provide useful insights to them.

7 Conclusion and future works

In this article, we show that using a *Deep Metric Learning* approach is a relevant solution for the fragments retrieval problem. We first produced baseline results making use of the *papyrus-level* information available (training directly on the *target-task*) with large amounts of data. We obtained our best results with the convolutional part of *VGG16* on the Hisfrag dataset, with a *top-1* accuracy of 0.87. On the Michigan dataset, the scores were lower due to a much lesser quantity of training data available. We then explored two ways in which such method can be used in a realistic scenario, trying to keep the human annotation work to a minimum. We showed that the *self-supervised* approach we propose outperforms the *domain adaptation* approach, while requiring no human annotation work and no pre-trained model, with a best *top-1* accuracy of 0.73 on the Hisfrag dataset. Finally, we experimented with using the *self-supervised* approach in a use case where papyrologists only have access to the unlabelled data they want to reconstruct. The results we obtained suggest that this approach should provide useful suggestions for papyrologists wanting to reconstruct new-found fragments, without having to manually annotate anything, depending on the properties and the quantity of data available.

In the future, we would like to conduct more thorough experiments on the impact that the patch selection process has on training the *self-supervised* models. Indeed, as [1] suggests, the whole process relies on the representativity of the patches chosen to represent the fragment, selecting informative patches within the fragment (regions that look the most similar or that look the most dissimilar) could help the *self-supervised* model learn more relevant features. The method for determining that two fragments are similar based on the scores computed on the patches could also be an interesting subject. Right now we use an average of the scores, but we could add a filtering phase to take outliers into account or imagine more sophisticated algorithms. Similarly, it would be interesting to work on some kind of iterative process to update our rankings based on the coherence of the associations, as new predictions are available, and possibly by taking into account expert input. We are also interested in automatic batch-level data augmentation at training time on challenging samples in order to improve training when little data is available.

[5] http://www.papyrologie.paris-sorbonne.fr/menu1/jouguet.htm

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Bondi, L., Güera, D., Baroffio, L., Bestagini, P., Delp, E.J., Tubaro, S.: A preliminary study on convolutional neural networks for camera model identification. Electronic Imaging **2017**(7), 67–76 (2017)
2. Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., Shah, R.: Signature verification using a siamese time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **7**(04), 669–688 (1993)
3. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 135–150 (2018)
4. Christlein, V., Gropp, M., Fiel, S., Maier, A.: Unsupervised feature learning for writer identification and writer retrieval. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 991–997. IEEE (2017)
5. Christlein, V., Nicolaou, A., Seuret, M., Stutzmann, D., Maier, A.: Icdar 2019 competition on image retrieval for historical handwritten documents (2019)
6. Cloppet, F., Eglin, V., Helias-Baron, M., Kieu, C., Vincent, N., Stutzmann, D.: Icdar2017 competition on the classification of medieval handwritings in latin script. In:

2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1371–1376. IEEE (2017)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
8. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision, pp. 1422–1430 (2015)
9. Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE transactions on pattern analysis and machine intelligence **38**(9), 1734–1747 (2015)
10. Fiel, S., Kleber, F., Diem, M., Christlein, V., Louloudis, G., Nikos, S., Gatos, B.: Icdar2017 competition on historical document writer identification (historical-wi). In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1377–1382. IEEE (2017)
11. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp. 1735–1742. IEEE (2006)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015). URL `http://arxiv.org/abs/1512.03385`
13. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, pp. 84–92. Springer (2015)
14. Huh, M., Agrawal, P., Efros, A.A.: What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614 (2016)
15. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis **6**(5), 429–449 (2002)
16. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
17. Karpinski, R., Belaid, A.: Semi-supervised learning through adversary networks for baseline detection. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 5, pp. 128–133. IEEE (2019)
18. KAYA M.; BLGE, H.: Deep metric learning: A survey. Symmetry (2019). DOI https://doi.org/10.3390/sym11091066
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
21. Korbar, B., Tran, D., Torresani, L.: Cooperative learning of audio and video models from self-supervised synchronization. arXiv preprint arXiv:1807.00230 (2018)
22. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European conference on computer vision, pp. 577–593. Springer (2016)
23. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 730–734 (2015). DOI 10.1109/ACPR.2015.7486599
24. Lombardi, F., Marinai, S.: Deep learning for historical document analysis and recognitiona survey. Journal of Imaging **6**(10), 110 (2020)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
26. Mahendran, A., Thewlis, J., Vedaldi, A.: Cross pixel optical-flow similarity for self-supervised learning. In: Asian Conference on Computer Vision, pp. 99–116. Springer (2018)
27. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Natural Language Engineering **16**(1), 100–103 (2008)
28. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6707–6717 (2020)
29. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: European Conference on Computer Vision, pp. 527–544. Springer (2016)
30. Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9359–9367 (2018)
31. Ostertag, C., Beurton-Aimar, M.: Matching ostraca fragments using a siamese neural network. Pattern Recognition Letters **131**, 336–340 (2020)
32. Owens, A., Efros, A.A.: Audio-visual scene analysis with self-supervised multisensory features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 631–648 (2018)
33. Paixo, T.M., Berriel, R.F., Boeres, M.C., Koerich, A.L., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Self-supervised deep reconstruction of mixed strip-shredded text documents. Pattern Recognition **107**, 107535 (2020). DOI https://doi.org/10.1016/j.patcog.2020.107535. URL `http://www.sciencedirect.com/science/article/pii/S0031320320303381`
34. Pal, S., Datta, A., Majumder, D.D.: Computer recognition of vowel sounds using a self-supervised learning algorithm. J. Anatomical Soc. India **6**, 117–123 (1978)
35. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
36. Pirrone, A., Beurton-Aimar, M., Journet, N.: Papy-s-net: A siamese network to match papyrus fragments. In: Proceedings of the 5th International Workshop on Historical Document Imaging and Processing, pp. 78–83 (2019)
37. Ren, Z., Lee, Y.J.: Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 762–771 (2018)
38. Saito T, R.M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS One (2015). DOI doi:10.1371/journal.pone.0118432
39. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
40. Seuret, M., Nicolaou, A., Stutzmann, D., Maier, A., Christlein, V.: Icfhr 2020 competition on image retrieval for historical handwritten fragments. International Conference on Frontiers of Handwriting Recognition (September, 2020). DOI 10.1109/ICFHR2020.2020.00048

41. Studer, L., Alberti, M., Pondenkandath, V., Goktepe, P., Kolonko, T., Fischer, A., Liwicki, M., Ingold, R.: A comprehensive study of imagenet pre-training for historical document image analysis. CoRR **abs/1905.09113** (2019). URL `http://arxiv.org/abs/1905.09113`
42. Tang, H., Zhao, Y., Lu, H.: Unsupervised person re-identification with iterative self-supervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0 (2019)
43. Tang, Y., Peng, L., Xu, Q., Wang, Y., Furuhata, A.: Cnn based transfer learning for historical chinese character recognition. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 25–29. IEEE (2016)
44. Wiggers, K.L., Junior, A.d.S.B., Koerich, A.L., Heutte, L., de Oliveira, L.E.S.: Deep learning approaches for image retrieval and pattern spotting in ancient documents. arXiv preprint arXiv:1907.09404 (2019)
45. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision, pp. 818–833. Springer (2014)
46. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision, pp. 649–666. Springer (2016)