IsOBS: An Information System for Oracle Bone Script

Xu Han[*] **, Yuzhuo Bai**[*]**, Keyue Qiu**[*]**, Zhiyuan Liu**[†] **, Maosong Sun**
State Key Lab on Intelligent Technology and Systems,
Institute for Artificial Intelligence,
Department of Computer Science and Technology, Tsinghua University, Beijing, China
{hanxu17,byz18,qky18}@mails.tsinghua.edu.cn
{liuzy,sms}@mail.tsinghua.edu.cn

Abstract

Oracle bone script (OBS) is the earliest known ancient Chinese writing system and the ancestor of modern Chinese. As the Chinese writing system is the oldest continuously-used system in the world, the study of OBS plays an important role in both linguistic and historical research. In order to utilize advanced machine learning methods to automatically process OBS, we construct an information system for OBS (IsOBS) to symbolize, serialize, and store OBS data at the character-level, based on efficient databases and retrieval modules. Moreover, we also apply few-shot learning methods to build an effective OBS character recognition module, which can recognize a large number of OBS characters (especially those characters with a handful of examples) and make the system easy to use. The demo system of IsOBS can be found from http://isobs.thunlp.org/. In the future, we will add more OBS data to the system, and hopefully our IsOBS can support further efforts in automatically processing OBS and advance the scientific progress in this field.

1 Introduction

Oracle bone script (OBS) refers to characters carved on animal bones or turtle plastrons. To research OBS is important for both Chinese linguistic and historical research: (1) As shown in Figure 1, OBS is the direct ancestor of modern Chinese and closely related to other languages in East Asia (Xueqin, 2002). Analysis and understanding of OBS is vital for studying the etymology and historical evolution of Chinese as well as other East Asian languages. (2) As shown in Figure 2, on one OBS document carved on one animal bone or turtle plastron, the number of characters ranges from fewer than ten to more than one hundred. Besides, as OBS is used for divination in ancient China, these documents cover a variety of topics, including war, ceremonial sacrifice, agriculture, as well as births, illnesses, and deaths of royal members (Flad et al., 2008). Therefore, OBS documents constitute the earliest Chinese textual corpora, and to analyze and understand OBS is of great significance to historical research.

Considering that it is often sophisticated and time-consuming to manually process ancient languages, some efforts have been devoted to utilizing machine learning techniques in this field. In order to detect and recognize ancient characters, Anderson and Levoy (2002); Rothacker et al. (2015); Mousavi and Lyashenko (2017); Rahma et al. (2017); Yamauchi et al. (2018) utilize computer vision techniques to visualize Cuneiform tablets and recognize Cuneiform characters, Franken and van Gemert (2013); Nederhof (2015); Iglesias-Franjo and Vilares (2016) apply similar techniques to recognize Egyptian hieroglyphs. For understanding the ancient text, Snyder et al. (2010) first show the feasibility of automatically deciphering a dead language by designing a Bayesian model to match the alphabet with non-parallel data. Then, Berg-Kirkpatrick and Klein (2011) propose a more effective decipherment approach and achieve promising results. Pourdamghani and Knight (2017) adopt a method similar to non-parallel machine translation (Mukherjee et al., 2018; Lample et al., 2018) to decipher related languages, which further inspires Luo et al. (2019) to propose a novel neural approach for automatic decipherment of Ugaritic and Linear B. Doostmohammadi and Nassajian (2019); Bernier-Colborne et al. (2019) explore to learn language models for Cuneiform Text.

These previous efforts have inspired us to apply machine learning methods to the task of processing OBS. However, there are still three main challenges:

[*] indicates equal contribution
[†] Corresponding author

Proceedings of the 2020 EMNLP (Systems Demonstrations), pages 227–233
November 16-20, 2020.

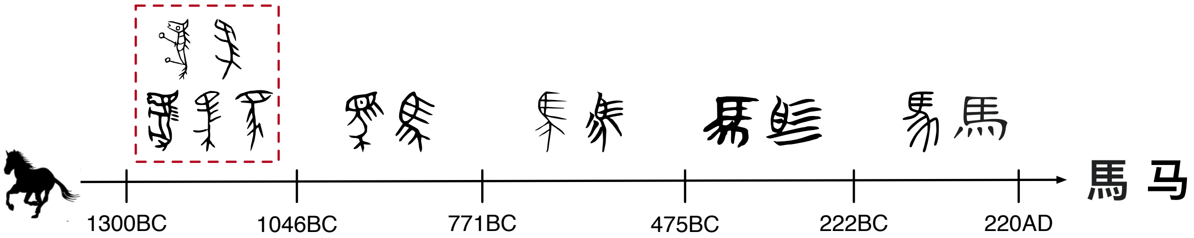


Figure 1: The historical evolution of the character "horse" from OBS to modern Chinese.

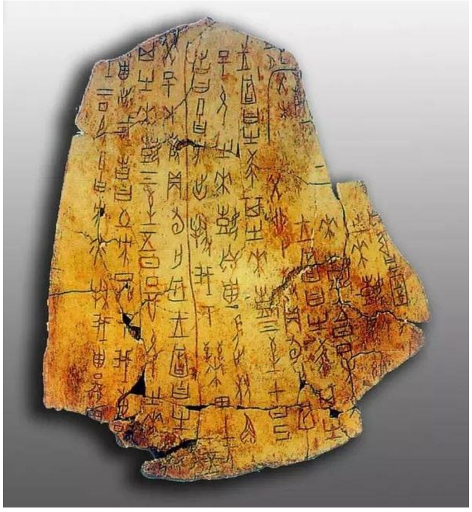


Figure 2: An example of an OBS document used in divination.

(1) Different from those ancient Greek and Central Asian scripts, in which letters are mainly used to constitute words and sentences, OBS is hieroglyphic and does not have any delimiter to mark word boundaries. This challenge also exists in modern Chinese scenarios. (2) Although OBS is the origin of modern Chinese, it is quite different from modern Chinese characters. Typically, one OBS character may have different glyphs. Moreover, there are many compound OBS characters corresponding to multiple modern Chinese words. (3) There still lacks an effective and stable system to symbolize and serialize OBS data. Most OBS data is stored in the form of unserialized bone/plastron photos, which cannot support either recognizing characters or understanding text.

The above three challenges make it difficult to use existing machine learning methods for understanding OBS, and the third one is the most crucial. To this end, we construct an information system for OBS (IsOBS) to symbolize and serialize OBS data at the character-level, so that we can utilize machine learning methods to process OBS in the future: (1) We construct an OBS character database, where each character is matched to corresponding modern Chinese character (if it has been deciphered) and incorporates a variety of its glyphs. (2) We construct an OBS document database, which stores more than $5,000$ OBS documents. We also split the images of these documents into character images, and use these character images to construct both the OBS and corresponding modern Chinese character sequences for each document. (3) We also implement a character recognition module for OBS characters based on few-shot learning models, considering there are only a handful of examples for each OBS character. Based on the character recognition module, we construct an information retrieval module for searching in character and document databases.

The databases, character recognition module, and retrieval module of IsOBS provide an effective and efficient approach to symbolize, serialize, and store the data of OBS. We believe IsOBS can serve as a footstone to support further research (especially character recognition and language understranding) on automatically processing OBS in the future.

2 Application Scenarios

As mentioned before, IsOBS is designed for symbolizing, serializing, and storing the OBS data. Hence, the application scenarios of IsOBS mainly focus on constructing databases for both OBS characters and documents, as well as implementing character recognition and retrieval modules for data search.

2.1 Character Database for OBS

In IsOBS, we construct a database to store OBS characters. For each OBS character, both its corresponding modern Chinese character (just for those OBS characters that have been deciphered) and glyph set will be stored. As shown in Figure 3, users can input a modern Chinese character to

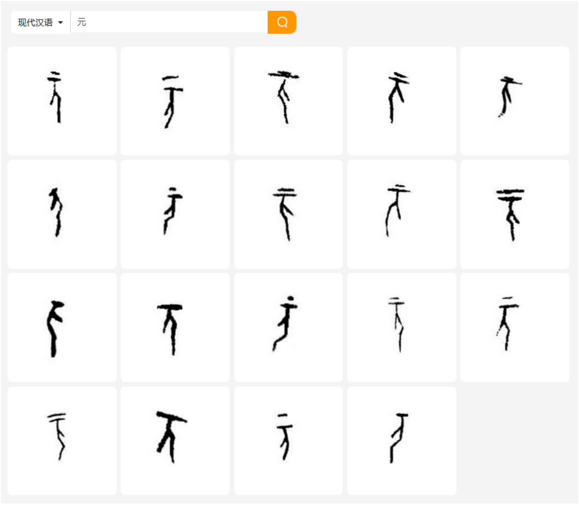
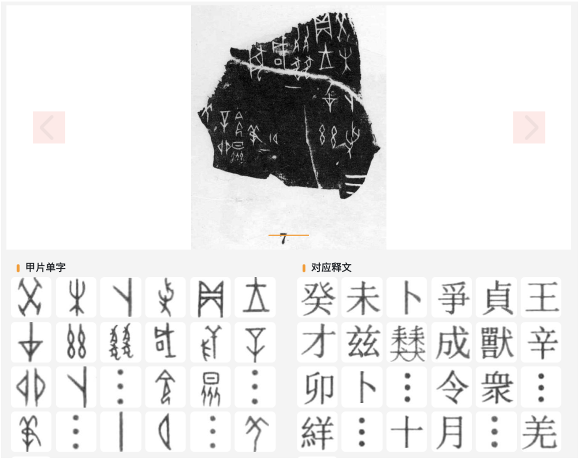


Figure 3: The example of the character database in IsOBS. Users input a modern Chinese character and get its corresponding 19 glyphs.



癸	未	卜	爭	貞	王
才	玆	𢀛	成	獸	辛
卯	卜	：	令	衆	：
絴	：	十	月	：	羌

Figure 4: The example of the document database in IsOBS. Users input an identity number and get its corresponding document.

search for all glyphs of its corresponding OBS character. For those OBS characters that have no corresponding modern Chinese characters, we provide interfaces to utilize our character recognition module to search them. We will later introduce this part in more details.

2.2 Document Database for OBS

Besides the character database, we also construct a document database to store OBS documents. As shown in Figure 4, for each document in the document database, we store the image of its original animal bones or turtle plastrons, and both the OBS and modern Chinese character sequences of this document. By querying the specific identity number designated by official collections, users can retrieve the corresponding OBS document from our database. In addition, we also align the character database with the document database, thus when users input one modern Chinese character to retrieve OBS glyphs, the documents mentioning this character can also be retrieved.

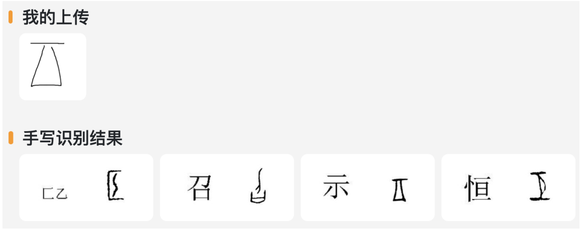


Figure 5: The example of the character recognition module in IsOBS. Users either write by hand or upload a glyph and get possible matches.

2.3 Character Recognition and Information Retrieval Modules

Since OBS characters are hieroglyphs and the character-glyph mappings are quite complex, the character recognition module is thus designed to deal with these complex mappings of input glyph images to their OBS characters. As shown in Figure 5, after we input the handwritten glyph image of the character, the character recognition module returns several latent matching pairs of OBS characters and their corresponding modern Chinese characters. Users can select one matching result for the next search. We also provide other commonly used retrieval methods (e.g. index retrieval), which is helpful for users to quickly find characters and documents in our system to conduct further research.

3 System Framework and Details

In this section, we mainly focus on introducing the overall framework and details of our system, especially introducing how to construct OBS databases and build the character recognition module. The overall framework of IsOBS including all databases and modules is shown in Figure 6.

3.1 OBS Databases

Our databases are constructed from two well-known collections. One is the collection of OBS rubbings and standardized characters compiled by experts in Chinese Academy of Social Sciences (CASS) (Moruo and Houxuan, 1982), and the

Name	Number of Classes	Number of Samples	Description
oracle300	353	11586	Classes with more than 20 examples
oracle600	617	15638	Classes with more than 12 examples
oracle1600	1621	20420	Classes with more than 2 examples

Table 1: The statistics of different dataset with different character sets.

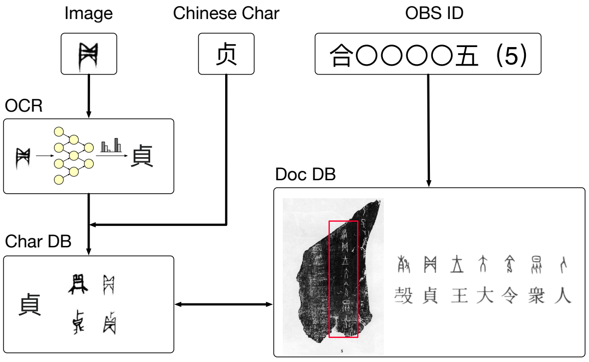


Figure 6: The overall framework of IsOBS including all databases and modules.

other one is the collection of variant written forms (glyphs) of OBS characters with their corresponding modern Chinese characters (Zhao et al., 2009).

For standardized OBS document collection, our databases now contain more than $5,000$ items, each including images of OBS rubbings, corresponding standardized OBS characters and their modern Chinese characters. Previous database platforms have not been able to cut out individual characters, making it difficult to support automatic operations. While our platform can provide finer-grained oracle data in a sequential form, which makes it easier for various electronic systems to conduct operations.

For hand-written OBS character collection, we obtain $22,161$ oracle character examples in $2,342$ classes, from which we create our dataset for training and testing our character recognition module.

3.2 Character Recognition Module

In available OBS character data, each character class usually has just a handful of examples. Due to the scarcity of OBS data, we adopt few-shot learning model for our classifier to capture the patterns from small amounts of data. Specifically, we implement prototypical network (Snell et al., 2017) for classification, which learns a non-linear mapping to embed examples into a feature space where those examples of the same class will cluster around a single prototype representation, as shown in Figure 7.

The architecture of the prototypical network is shown in Figure 8, and we denote the prototypical network as $f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$ for simplicity, where ϕ is the parameters to be learned by training, D and M stand for the dimension of the input data and the dimension of the embedded features respectively.

For each class, the prototype c_i is set as the average of the embeddings of the support set, so the prototype of the class i can be denoted as $c_i = \frac{1}{n_i}\sum_{j=1}^{n_i} f_\phi(x)$, where n_i is the number of samples in the support set of the class.

For each query x, we use f_ϕ to embed the query instance, then compute the distribution of x by the softmax of euclidean distances between $f_\phi(x)$ and the prototypes of each class, in other words,

$$p_\phi(y = i|x) = \frac{\exp(-d(f_\phi(x), c_k))}{\sum_{i'} \exp(-d(f_\phi(x), c_{i'}))}.$$

Aside from prototypical network, we apply other neural networks for comparison, and finally select the most powerful one for our character recognition module. We adopt relation network (Sung et al., 2018), which is also an effective model in the area of few-shot learning, and siamese network (Chopra et al., 2005), for it is also a widely-used model in the area of character classification.

4 Experiment and Evaluation

We evaluate different character recognition models on self-created dataset. The results show that our implementation of prototypical network can achieve stable and competitive results. The datasets and source code can be found from `https://github.com/thunlp/isobs`.

4.1 Dataset

Our newly created dataset is obtained from the collection of hand-written OBS characters mentioned in 3.1. The whole dataset contains $22,161$ character images from $2,342$ classes annotated by experts in OBS character research, each class refers to a unique character and is available on our website.

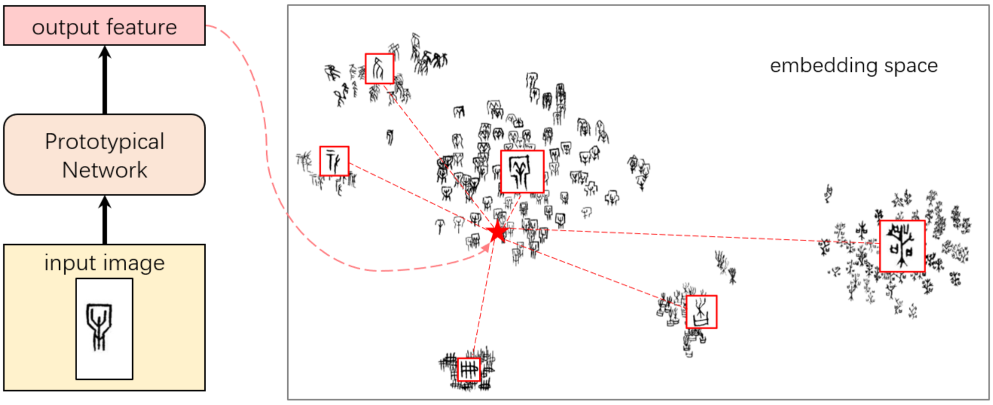


Figure 7: Illustration of prototypical network, with the glyph coordinates in space drawn by t-SNE according to $f_\phi(x)$.

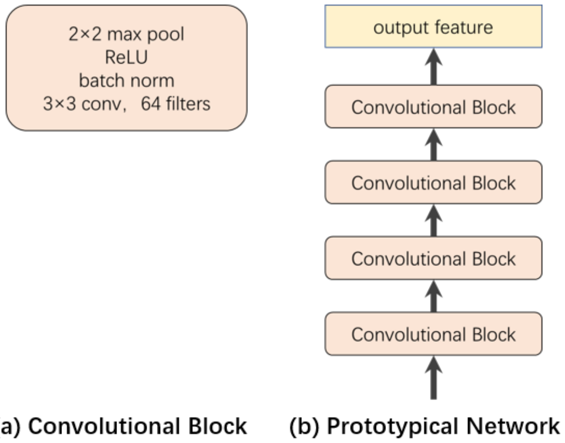


Figure 8: The architecture of prototypical network.

Each image in the dataset is 110 by 200. Considering that both the training and test set should not be empty for each class, our experiment is conducted on part of the dataset, which contains $1,621$ classes and $20,420$ character images. Due to the lack of enough few-shot training data for certain classes, we created three datasets as shown in Table 1. Each dataset is partitioned into training examples and test examples in 3 or 4 to 1 ratio.

4.2 Evaluation Metric

As mentioned above, we use prototypical network to classify OBS characters. For the training part, we use typical few-shot learning method to train the prototypical network. For the evaluation part, as aiming to evaluate the practicability of the model as an OBS character classifier, we score our model by using the top-k accuracy of the whole classification over given dataset, rather than common few-shot learning evaluation. Considering that only the classes in oracle300 have ample data to do few-shot training, we use the training set of oracle300 to train our model, and perform classification evaluation respectively on oracle300, oracle600 and oracle1600.

4.3 Neural Network Hyper-Parameters

For the few-shot learning models, in each epoch, we train 100 steps. In each step, we randomly select 60 classes for training prototypical network, while the number of selected classes for relation network is 5. For each class, there are 5 randomly chosen support examples and 5 query examples. The learning rate is set to 0.001 at the beginning, and decreases by half for every 20 (for prototypical network) or $100,000$ (for relation network) steps. For siamese network, the learning rate is set to 0.0001, and weight-decay 0.00001.

4.4 Overall Results

Table 2 shows the overall performance of prototypical network on different datasets, and Table 3 shows the performance of different models on oracle600. From these two tables, we can find that:

(1) Prototypical network performs well on both oracle300 and oracle600, with the top-10 accuracy more than 90%.

(2) When generalized to oracle1600, which is larger and consists classes that contains scanty examples, our model still reaches 54.4% accuracy, indicating that our model works in generalized circumstance. As we just train models on oracle300 i.e, most characters in the test sets are not contained in the training set, this is a quite difficult scenario.

(3) Prototypical network notably outperforms

Dataset	hit@1	hit@3	hit@5	hit@10
oracle300	69.4	84.1	88.1	92.3
oracle600	66.0	80.7	85.1	90.0
oracle1600	54.4	69.1	73.8	78.4

Table 2: The result for prototypical network on different dataset (%).

Model	hit@1	hit@5	hit@10
Siamese Network	6.1	16.1	25.2
Relation Network	18.1	45.1	57.7
Prototypical Network	66.0	85.1	90.0

Table 3: The result for prototypical network, relation network, and siamese network on oracle600 (%).

the other two models, which might result from the high-efficiency of few-shot learning on training set with sparse examples and massive classes, as well as the transportability of prototypical network to character classification tasks. Compared to prototypical network, siamese network uses all the training set to train models, which makes it hard to converge; relation network works well on training, but its concatenation and relation modules make it difficult to transfer from few-shot learning to character classification task where the number of examples in each class varies, so the utilization rate of the extra examples is low.

Considering prototypical network outperforms other models, our character recognition module is finally based on prototypical network.

5 Conclusion and Future Work

As to research OBS is important for both Chinese linguistic and historical research, we thus construct an information system for OBS and name the system IsOBS. IsOBS provides an open digitalized platform consisting of the OBS databases, the character recognition module, and the retrieval module. The experimental results further demonstrate that our character recognition module based on few-shot learning models have achieved satisfactory performance on our self-created hand-written OBS character dataset.

In the future, we plan to explore the following directions: (1) to include more OBS document and character data from collection books into our existing databases, (2) to employ generative learning and adversarial algorithms to add more robustness to our model, and (3) to construct a language model for ancient languages. We believe that these three directions will be beneficial for ancient languages research and support further exploration of utilizing machine learning for understanding OBS.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503), the National Natural Science Foundation of China (NSFC No. 61732008) and Beijing Academy of Artificial Intelligence (BAAI). Bai and Qiu are supported by Tsinghua University Initiative Scientific Research Program.

References

Sean E. Anderson and Marc Levoy. 2002. Unwrapping and visualizing cuneiform tablets. *IEEE Computer Graphics and Applications*, 22(6):82–88.

Taylor Berg-Kirkpatrick and Dan Klein. 2011. Simple effective decipherment via combinatorial optimization. In *Proceedings of EMNLP*, pages 313–321.

Gabriel Bernier-Colborne, Cyril Goutte, and Serge Léger. 2019. Improving cuneiform language identification with bert. In *Proceedings of VarDial*, pages 17–25.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of CVPR*, pages 539–546.

Ehsan Doostmohammadi and Minoo Nassajian. 2019. Investigating machine learning methods for language and dialect identification of cuneiform texts. *Proceedings of NAACL-HLT*, pages 188–193.

Rowan K Flad, Sarah Allan, Rod Campbell, Xingcan Chen, Lothar von Falkenhausen, Hui Fang, Magnus Fiskesjö, Zhichun Jing, David N Keightley, Evangelos Kyriakidis, et al. 2008. Divination and power: a multiregional view of the development of oracle bone divination in early china. *Current Anthropology*, 49(3):403–437.

Morris Franken and Jan C van Gemert. 2013. Automatic egyptian hieroglyph recognition by retrieving images as texts. In *Proceedings of MM*, pages 765–768.

Estíbaliz Iglesias-Franjo and Jesús Vilares. 2016. Searching four-millennia-old documents: A text retrieval system for egyptologists. In *Proceedings of LaTeCH*, pages 22–31.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of ICLR*.

Jiaming Luo, Yuan Cao, and Regina Barzilay. 2019. Neural decipherment via minimum-cost flow: From ugaritic to linear b. In *Proceedings of ACL*, pages 3146–3155.

Guo Moruo and Hu Houxuan. 1982. *The collection of Oracle Bone scripts*.

Seyed Muhammad Hossein Mousavi and Vyacheslav Lyashenko. 2017. Extracting old persian cuneiform font out of noisy images (handwritten or inscription). In *Proceedings of MVIP*, pages 241–246.

Tanmoy Mukherjee, Makoto Yamada, and Timothy Hospedales. 2018. Learning unsupervised word translations without adversaries. In *Proceedings of EMNLP*, pages 627–632.

Mark-Jan Nederhof. 2015. Ocr of handwritten transcriptions of ancient egyptian hieroglyphic text. *Altertumswissenschaften in a Digital Age: Egyptology, Papyrology and beyond, Leipzig*.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of EMNLP*, pages 2513–2518.

Abdul Monem S Rahma, Ali Adel Saeid, and Muhsen J Abdul Hussien. 2017. Recognize assyrian cuneiform characters by virtual dataset. In *Proceedings of ICTA*, pages 1–7.

Leonard Rothacker, Denis Fisseler, Gerfrid GW Müller, Frank Weichert, and Gernot A Fink. 2015. Retrieving cuneiform structures in a segmentation-free word spotting framework. In *Proceedings of HIP*, pages 129–136.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of NIPS*, pages 4077–4087.

Benjamin Snyder, Regina Barzilay, and Kevin Knight. 2010. A statistical model for lost language decipherment. In *Proceedings of ACL*, pages 1048–1057.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of CVPR*, pages 1199–1208.

Li Xueqin. 2002. The xia-shang-zhou chronology project: methodology and results. *East Asian Archaeology*, 4(1):321–333.

Kenji Yamauchi, Hajime Yamamoto, and Wakaha Mori. 2018. Building a handwritten cuneiform character imageset. In *Proceedings of LREC 2018*.

Liu Zhao, Hong Biao, and Zhang Xinjun. 2009. *The new collection of Oracle Bone scripts*.