# Bessarion: Medieval Greek Inscriptions on a challenging dataset for Vision and NLP tasks

Giorgos Sfikas[1] and Panagiotis Dimitrakopoulos[2] and George Retsinas[3] and Christophoros Nikou[2] and Pinelopi Kitsiou[4]

[1] Department of Surveying and Geoinformatics Engineering
University of West Attica, Greece
[2] Department of Computer Science and Engineering
University of Ioannina, Greece
[3] School of Electrical and Computer Engineering
National Technical University of Athens, Greece
[4] Ephorate of Antiquities of Arta, Greece
gsfikas@uniwa.gr, p.dimitrakopoulos@uoi.gr, gretsinas@central.ntua.gr, cnikou@uoi.gr, kpinelopi@gmail.com

**Abstract.** We present a text and imaging dataset of Byzantine-era Medieval Greek inscriptions, suitable as a challenging testbed for Computer Vision and Natural Language Processing tasks. The lack of sizable related training sets, as well as difficulties related to the historical character and content of the inscriptions (natural wear of characters, systematic misspellings, etc.) make for a context where modern resource-hungry techniques are not straightforward to apply. We describe the dataset contents – images, geometric and text annotation, metadata – and discuss baselines for three Computer Vision tasks (Inscription Detection, Text Recognition) and one Natural Language Processing task (Word Classification). The dataset is publicly available at https://github.com/Archaeocomputers/Bessarion.

**Keywords:** Medieval Greek, Donative Inscriptions, Object Detection, Text Recognition, Word Classification, Text Classification, Text Categorization

## 1 Introduction

Bessarion was a scholar that lived during the twilight of the Roman empire in the $15^{th}$ century [3]. We have used his name for the dataset that we present in this work: A dataset that is made up of annotated images of *donative Byzantine* inscriptions. Let us explain the two terms, "donative" and "Byzantine": The term "Donative" refers to an inscription that informs us about who funded the construction of the site where the inscription is situated. The term "Byzantine" refers to inscriptions that have been written during the days of the late Roman empire and/or are closely related to the stylistic traits, character, or institutions of the late Roman state. We can see an example of dataset samples in Figure 1.

**Fig. 1.** Example images of the Bessarion dataset. Images depict historical donative Byzantine inscriptions, describing lists of the persons or groups that contributed for the construction of a related site or monument. The text is written in Greek.

We argue that the Bessarion dataset is useful for the Document Imaging and Natural Language Processing (NLP) community, as a challenging testbed for vision and language processing tasks. It is a dataset that represents considerable distribution shift [5] with respect to most existing datasets in several ways. First, the inscriptions are written in the medieval phase of the Greek language, and they employ a very special form of the Greek script. Both aspects are very little documented in data science applications; notable exceptions are [6], which focuses on using NLP techniques to support Handwritten Text Recognition (HTR), or [2], testing segmentation and HTR methods on collections of handwritten Byzantine text. Second, this is a constrained dataset in terms of resources. The available inscriptions are no more than a few dozen, and in total the character tokens do not sum up to more than a few thousand. Unlike other, resource-rich languages and scripts, for this language and script combination there is very little on which to pretrain or use as foundational basis for a vision, NLP or other learning model. Note that the stylistically closest data are handwritten Byzantine texts [2, 6] or text on Greek Papyri [8], which are still quite different in form compared to the material presented in this paper (see for example Fig. 2). In this respect, we hope that the Bessarion dataset will aid the community in elaborating new solutions for this new challenging application terrain.

The remainder of this paper is as follows. In Section 2 we outline the characteristics of the dataset in general. In Sections 3, 4, 5, we present data and

baselines that are related to three tasks, namely Inscription Detection, Text Recognition, and Word Classification. We close with general remarks and future work in Section 6.

## 2 Dataset outline

The current dataset contains NLP-related metadata for (part of) the included inscriptions. Donative Byzantine inscriptions contain orthographical imperfections up to a large extent, which, given contemporary conditions, are often surprising. In particular, the text systematically contains misspellings, many times even with different misspelled variants of the same word in the same inscription. It is therefore obvious that a "simple" natural language processing system will find insurmountable difficulties in analyzing the text, since the same word with the same meaning appears in a different way from the same "hand". The multitude of scribal errors adds an extra layer of difficulty to the task of natural language processing. The other major challenge is the small volume of the total text, since we have fully annotated inscriptions (i.e., with metadata with details about the full transcription and semantics including the identity of the site founder and the dating of the site) from a total of 25 inscriptions (see also Section 5 for more details).

**Table 1.** A table with numerical "facts" over the whole dataset.

| | |
|---|---|
| Total number of sites | 37 |
| Inscriptions with full metadata | 25 |
| Number of images | 122 |
| Outlined textlines | 504 |
| Outlined words | 2,776 |
| Outlined characters | 10,414 |

## 3 Inscription Detection task

In the case of text understanding applications, the primary goal is to detect regions containing only textual information, either as holistic region information or as textual parts at line, word or even character level. The detection problem, in the current case of identifying Byzantine "donative" inscriptions can be a challenging one, due to the variety of text appearance, the unconstrained locations of text within the natural image, degradations of text components over hundreds of years, as well as the overall complexity of each scene. While the majority of images in the dataset are inscription-centric, accurately detecting their boundaries presents several challenges. Lighting conditions can vary significantly between images, and factors like different viewpoints and writing styles can also pose difficulties for a detection method.


**Fig. 2.** An illustration of examples of the (Greek) letters found in the inscriptions of the dataset presented in this paper.

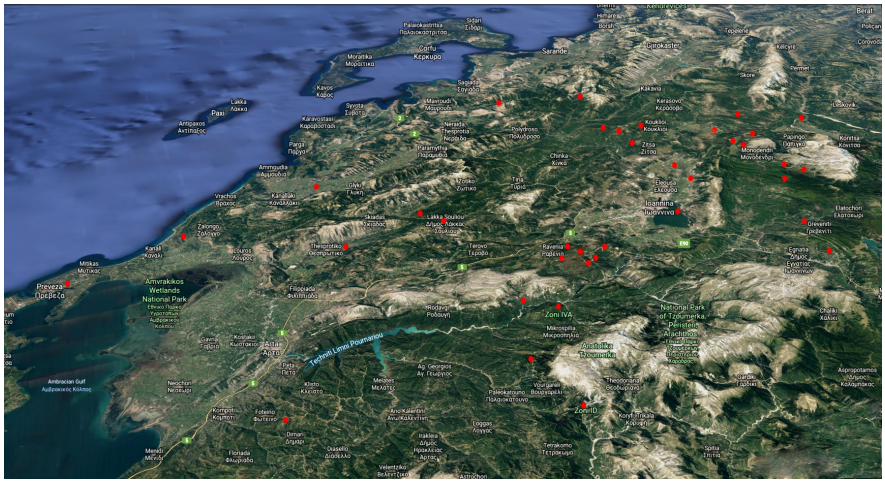

**Fig. 3.** Map showing positions of the sites relevant to *Bessarion* data. Our dataset comes from a total of 37 sites spanning the region of Epirus, situated in North-Western Greece.

### 3.1 Outline of data and annotation

The dataset contains in total 122 images of Byzantine inscriptions. Each image is object-centric, in the sense that it depicts a single donative inscription. All inscriptions are meticulously annotated with a bounding polygon that tightly encloses the text information. Fig. 4 depicts four inscriptions located in different Byzantine churches and the ground-truth polygon for each text region is illustrated with green color. One prominent characteristic is the almost rectangular shape which stays almost consistent in the whole dataset with some exceptions being evident (e.g. see top-right inscription of Figure 1).

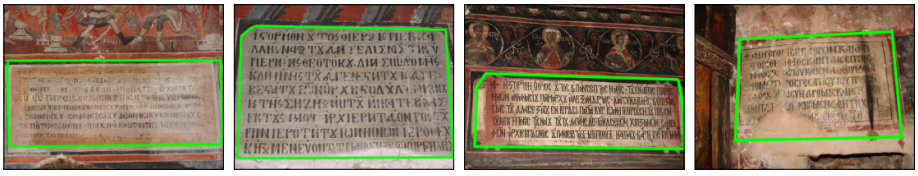

**Fig. 4.** Example ground-truth annotations for selected samples. Bounding boxes shown in green highlight the text regions.

### 3.2 Baseline methods

We are interested in detecting a rectangular-shaped polygon which includes the text in a wall-mounted inscription. We deploy two different methods in order to quantify their detection performance. These methods must not only address the challenges mentioned earlier but also function effectively with a limited dataset, which is small by deep learning standards. Additionally, an ideal detector for this scenario should be lightweight, with a small number of parameters, to facilitate possible deployment on nodes such as smartphones.

***Sparse R-CNN.*** We deployed the Sparse R-CNN detector, as described in [11]. This is a state-of-the-art two stage detector method, that first generates region proposals and in a second stage applies classification and localization. This method is characterized by a fixed set of learned object proposals, which are provided to the object recognition head to get bounding boxes. Finally, the model outputs predictions directly, without requiring a non-maximum suppression post-processing step, leading to faster inference time. The initial input sparse set of proposal boxes and features, together with the one-to-one dynamic instance interaction allow this method to thrive in our case where the dataset is considerably small with only one object class and up to a hundred training samples.

***Quaternion GANs.*** Furthermore we evaluated the performance of Quaternion Generative Adversarial Networks (Q-GANs) proposed in [9]. This method is a quaternionic adaptation of the well-known pair of the generator and discriminator networks that are used in standard GANs. The introduction of quaternion convolutional layers, which have quaternionic parameters and activations, leads to a reduction in the number of parameters. Quaternionic and hypercomplex models, apart from leading to better image classification results than traditional CNNs, have the property to treat RGB channels holistically, and not as three independent entities [10]. In contrast to traditional detection methods, the adversarial network treats text detection as a semantic segmentation task. This approach generates a binary output that indicates the presence of text. Bounding boxes are then extracted from this output using thresholding and maximum connected component analysis.

### 3.3 Numerical Comparison

To train and numerically evaluate the baseline methods we have chosen to partition the dataset to a training and test set according to a 80%/20% split. All images were then resized so as their width was at most 1024 pixels, while keeping their aspect ratio fixed. During training we introduced data augmentation which includes random rotations, zoom / cropping and translations.

In Table 2, we report numerical results in terms of mean average precision (mAP). Both methods achieve sufficiently good performance despite the changing detection setup. The Sparse R-CNN model slightly outperforms the Q-GAN but at the cost of significantly higher total number of parameters. Qualitative detection results are showed in Fig.5 for the Sparse R-CNN method. Despite

achieving acceptable results, both methods struggle to perfectly detect all inscriptions. This highlights the ongoing challenges: One specific challenge relates to the ambiguity in determining the precise location where the text ends. This ambiguity contributes to the lower numerical scores achieved by both methods.

**Table 2.** Numerical comparison of baseline detectors. Detection accuracy in terms of mean average precision and average precision at different IoU thresholds is reported. Network sizes are cited for comparison (counted in numbers of millions of parameters).

| Method | AP | $AP_{70}$ | $AP_{50}$ | Parameter Size |
|---|---|---|---|---|
| Sparse R-CNN | 0.56 | 0.82 | 0.63 | 105.94 M |
| Q-GAN + CC | 0.37 | 0.62 | 0.49 | 1.6 M |

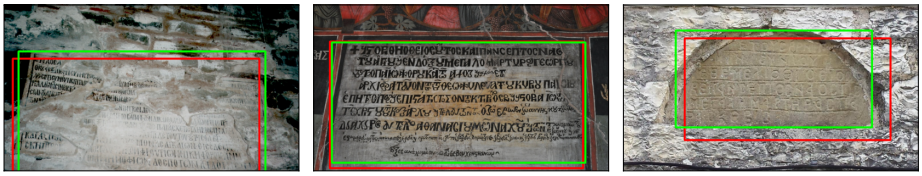

**Fig. 5.** Detection results for the Sparse R-CNN method. Ground truth inscription bounding boxes are shown in green color, while predicted ones are depicted in red.

**Fig. 6.** Examples of recognition challenges posed by the nature of Byzantine text painting. Two letters may appear in an unconventional relative position of one to the other, with the preceding letter written on top of the subsequent letter (for example the *tau* over the *omega* in the first image from the left), or forming a special complex (the *omicron* and *upsilon* in the second image), or "embracing it" (the *rho* over the *epsilon* in the third image), with letters in general not "respecting" the "convex hull" bounds of neighbouring letters (e.g. *epsilon* and *tau* in the fourth image).

## 4 Text Recognition task

Text Recognition is crucial for applications involving text image understanding, digitization, preservation, and accessibility of cultural heritage sites. Unlike

recognition of machine-printed text, handwriting is related to a number of unique characteristics that make the task much more challenging. In addition to the classical challenges, recognizing Byzantine text specifically poses further complexities. Text located on church walls introduces these additional challenges, which we will discuss in more detail. While Byzantine inscriptions often exhibit font-like characteristics, such as consistent letter forms and clear line formatting, the same format poses restrictions. Notably, there is no evident point where the words separate from each other. Furthermore, common stop words like "τoy" or "τvn" and several bi-characters (character complexes) can be written as one symbol thus posing severe limitations to character-to-character recognition systems (see Fig. 6). Additionally, Byzantine churches and monasteries that house these text inscriptions are many centuries old. Over time, it is natural that wall-mounted inscriptions degrade due to exposure to the elements.

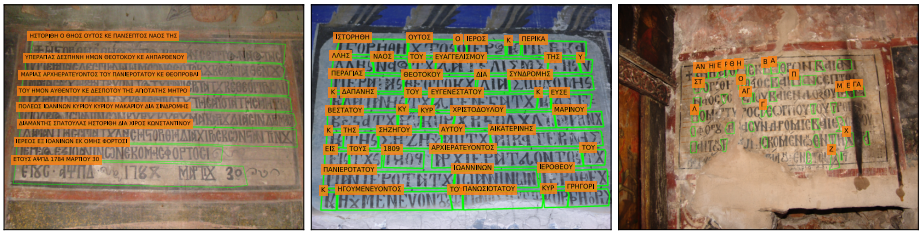

**Fig. 7.** Example of three different types of text annotations in the *Bessarion* dataset. From left to right: Line-level, word-level, character-level annotation.

### 4.1 Outline of data and annotation

The text level annotation of the *Bessarion* dataset can be categorized into three main types. The first category of annotations focuses on text lines. Each line in a particular inscription is localized using a polygon alongside with the corresponding text (see Fig. 7 left). Second, the dataset contains also word level annotations. Each word is meticulously annotated with a polygon, as shown in the middle panel of Fig. 7. It is important to note the limited spacing between words. Finally, some inscriptions are sparsely annotated in a letter/character level. This means that the boundaries of some letters are provided alongside their text ground truth, as illustrated again in Fig. 7. Table 3 also shows the overall statistics, including the number of different Byzantine inscriptions and individual annotations the dataset contains for each type of text annotation.

### 4.2 HTR Baseline Method

We deployed a Handwritten Text Recognition (HTR) system to accurately recognize Byzantine text from entire text lines. Specifically we run our experiments

**Table 3.** Number of individual annotation examples per annotation type. Number of different inscriptions where examples are extracted.

| Type | Lines | Words | Characters |
|---|---|---|---|
| Different Inscriptions | 23 | 19 | 14 |
| Individual Annotations | 193 | 1524 | 7552 |

with the HTR model proposed by Retsinas et al. [7]. This model accepts an image of either a word or a line of text as input and predicts the corresponding sequence of characters. The model follows a convolutional-recurrent architecture. This typically consists of a convolutional neural network (CNN) backbone for feature extraction, followed by a recurrent neural network (RNN) head for sequence modeling. The RNN component utilizes three stacked Bidirectional Long Short-Term Memory (BiLSTM) layers for efficient character recognition. The model is trained using a Connectionist Temporal Classification (CTC) loss function.

### 4.3 Numerical Evaluation

We partitioned the dataset to a 80%/20% training / test split. All input images are pre-processed by resizing them to a fixed resolution while maintaining their original aspect ratio. The training of the HTR system is performed via an Adam optimizer using an initial learning rate of $10^{-3}$ which gradually decreases using a multi-step scheduler. Because of the limited amount of training lines and the challenging setup of the Byzantine text, we used also word-level and character-level annotations to serve as data augmentation. Introducing individual words to the training set significantly boosted performance by particularly aiding the method to recognize the space between words in an given text line.

In Table 4, we summarize the numerical performance of the HTR method in terms of Character Error Rate (CER) and Word Error Rate (WER). We can see that the introduction of word and character level annotations in the training set vastly improved the recognition performance. Furthermore, in Fig. 8 we plot six text lines from the test set along with their corresponding ground truth text annotations and the HTR model's predicted text. Despite the limited training data, the method is able to yield predictions that are quite close to the original text.

## 5 Word Classification

In this section, we explore a Natural Language Processing task in the context of Byzantine "donative" inscriptions. This is a text-related task, and as in the imaging tasks there are characteristics that pose considerable difficulty here as well. Challenges include the use of a medieval Greek script and language on equally old and weathered wall-mounted inscriptions. We are interested in answering

**Table 4.** Numerical evaluation of Retsinas et al. HTR method [7] trained with different types of annotations. Test character error rate (CER) and word error rate (WER) are reported (lower values are better).

| Text annotations used | CER | WER |
|---|---|---|
| Text Lines | 0.564 | 0.903 |
| Text Lines + Words | 0.021 | 0.066 |
| Text Lines + Words + Characters | 0.021 | 0.066 |


**Fig. 8.** Example text lines with the ground truth recognition label *(Original)* and the HTR model one as (*Predicted*).

mainly two types of questions: a) which person donated or contributed for the specific monument, b) when was the monument constructed. We choose these questions, as they cover the main content that defines donative inscriptions.

Concerning our baseline NLP method, we aim to answer the aforementioned questions via a word classfication task. As medieval Greek is poorly represented in terms of accessible digitized corpora, we combine a BERT encoder-based model [1] with question-specific corpus augmentation methods.

### 5.1 The form of the ground truth

For each of the labels for which we have a complete notational ground truth (i.e., in the sense of having complete metadata information), we have information about the semantic role of each word present in the label in the form of a JSON structure. In figure 9 we can see an example of such a structure. We wish to build a system that will automatically evaluate a number of these fields, given the text of the inscription. Specifically, we need as a result:

- The words or numbers mentioned in the dating (ground truth in the fields: *date_intext*, *year_words*, *month_words*)
- The way of recording the date (ground truth in the field: *date_type*)
- The words that refer to the founder or founders (ground truth in the field: $founder\_intext$)

At the same time, this information will be necessary during the training of the system we will build. Additionally, we will also need information from the *name_words* field, which records words that are first names, without necessarily

```
{
    "site_id": "13",
    "name": "Αγία Φανερωμένη",
    "village": "Φορτόσι Κατσανοχωρίων",
    "date": "23 Οκτωβρίου 1787",
    "founder": "Παναγιώτης Βενέτης (Συνδρομή και επιστασία) και οι αδελφοί Κωνσταντίνος και Γεώργιος Λιανός (Συνδρομή και δαπάνη)",
    "inscriptions": [
        {
            "filename": "Agia Fanerwmeni - Katsanoxwria - Fortosi/20150315_144055.jpg",
            "text": "ΗΣΤΟΡΙΘΙ Ο ΘΙΟΣ ΟΥΤΟΣ ΚΕ ΠΑΝΣΕΠΤΟΣ ΝΑΟΣ Τ ΥΠΕΡΑΓΙΑΣ \n ΔΕΣΠΗΝΗΣ ΗΜΩΝ ΘΕΟΤΟΚΟΥ ΚΕ ΑΕΙ ΠΑΡΘΕΝΟΥ ΜΑΡΙΑΣ ΑΡΧΗΕΡΑΤΕΥΟΝΤΟΣ
            ΤΟΥ ΠΑΝΙ \n ΕΡΟΤΑΤΟΥ ΚΕ ΛΟΓΙΟΤΑΤΟΥ ΜΙΤΡΟΠΟΛΙΤΟΥ Τ ΑΓΙΟΤΑΤΗΣ ΜΙΤΡΟΠΟΛΕΟΣ ΙΩΑΝΙΝΩΝ ΚΥΡΙΟΥ ΚΥΡΙΟΥ \n ΜΑΚΑΡΙΟΥ ΔΙΑ ΣΗΝΔΡΟΜΗΣ ΚΕ
            ΕΠΙΤΡΟΠΙΑΣ ΑΠΟ ΤΟΝ ΠΑΝΤΟΚΡΑΤΟΡΑΝ ΚΕ ΑΝΟΘΕΝ ΕΝΕΑ ΚΟΥΜΠΕΔΕΣ ΠΑΝΑ \n ΓΙΟΤΟΥ ΒΕΝΕΤΙ Ο ΠΑΝΤΟΚΡΑΤΟΡ ΚΕ ΘΕ ΜΑΝΟΥΗΛ ΔΙΑ ΧΗΡΟΣ ΝΙΚΟΛΑΟΥ
            ΠΛΑΚΥΔΑ Ο ΔΕ ΠΡΟΔΡΟΜΟΣ Κ Η ΠΑΝΑΓΙΑ ΔΙΑ ΧΙΡΟΣ \n ΑΘΑΝΑΣΙΟΥ ΤΟΥ ΙΟΥ ΑΥΤΟΥ ΤΟ ΔΕ ΙΕΡΟΝ ΚΕ ΔΙΟ ΚΟΥΜΠΕΔΕΣ ΕΞΟ ΤΟΥ ΙΕΡΟΥ ΔΙΑ ΧΗΡΟΣ
            ΧΡΙΣΤΟΥ ΙΕΡΕΟΣ ΚΕ ΓΕΟΡΓΙΟΥ ΤΩΝ ΑΥΤΑΤΑΔΕΛΦΩΝ \n ΑΠΟ ΤΟΝ ΜΕΓΑΛΗΣ ΒΟΥΛΗΣ ΑΓΓΕΛΩΝ ΚΕ ΑΠΟΚΑΛΙΨΙΝ ΚΕ ΠΑΣΑ ΠΝΟΗΝ Κ ΚΑΤΟΘΕΝ Η ΕΞΙ
            ΚΟΥΜΠΕΔΕΣ ΔΙΑ ΣΗΝΔΡΟΜΗΣ ΚΕ ΔΑ \n ΠΑΝΗΣ ΚΩΝΣΤΑΝΤΙΝΟΥ ΚΕ ΓΕΩΡΓΙΟΥ ΤΩΝ ΛΙΑΝΩΝ ΚΑΙ ΔΙΑ ΧΗΡΟΣ ΚΑΤΟΘΕΝ ΕΞΙ ΚΟΥΜΠΕΔΕΣ ΚΩΝΣΤΑΝΤΙΝΟΥ
            ΙΕΡΕ \n ΟΣ ΕΚ Τ ΧΟΡΑΣ ΤΑΥΤΗΣ ΚΕ ΣΤΕΡΙΟΥ ΜΑΘΙΤΟΥ ΑΥΑΤΟΥ ΑΨΠΖ 1787 ΟΚΤΟΜΒΡΙΟΥ 23"
        }
    ],
    "comment": "Ο ναός τοιχογραφήθηκε με δαπάνη διαφορετικών κτιτόρων και αυτό σημειώνεται στην επιγραφή. Αναφέρεται ποια τμήματα του ναού
    διακοσμήθηκαν από διαφορετικούς ζωγράφους με τις αντίστοιχες χορηγίες.",
    "date_intext": "ΑΨΠΖ 1787 ΟΚΤΟΜΒΡΙΟΥ 23",
    "year_words": "ΑΨΠΖ 1787",
    "month_words": "ΟΚΤΟΜΒΡΙΟΥ",
    "date_type": "AnnoDomini",
    "founder_intext_extended": "ΔΙΑ ΣΗΝΔΡΟΜΗΣ ΚΕ ΕΠΙΤΡΟΠΙΑΣ ΑΠΟ ΤΟΝ ΠΑΝΤΟΚΡΑΤΟΡΑΝ ΚΕ ΑΝΟΘΕΝ ΕΝΕΑ ΚΟΥΜΠΕΔΕΣ ΠΑΝΑΓΙΟΤΟΥ ΒΕΝΕΤΙ ΔΙΑ ΣΗΝΔΡΟΜΗΣ
    ΚΕ ΔΑΠΑΝΗΣ ΚΩΝΣΤΑΝΤΙΝΟΥ ΚΕ ΓΕΩΡΓΙΟΥ ΤΩΝ ΛΙΑΝΩΝ",
    "founder_intext": "ΒΕΝΕΤΙ ΚΩΝΣΤΑΝΤΙΝΟΥ ΓΕΩΡΓΙΟΥ ΛΙΑΝΩΝ",
    "name_words": "ΠΑΝΑΓΙΟΤΟΥ ΒΕΝΕΤΙ ΚΩΝΣΤΑΝΤΙΝΟΥ ΓΕΩΡΓΙΟΥ"
},
```


**Fig. 9.** Sample ground-truth file in JSON format.

being all builders (for example, the name of the saint to whom the temple is dedicated, or the patronymic of a builder).

### 5.2 Processing Pipeline

The processing line that we recommend consists of the following sections:

1. Encoding each recognized word as a vector with semantic / contextual load.
2. Processing vectors with a Neural Network.
3. Categorization of each word based on its role in the building inscription.

**Encoding.** We use the GreekBERT encoder as a semantic feature extractor [4]. In a first phase, the text is analyzed into small components (tokens) based on a vocabulary of possible elements. In the simplest version of the process, we can understand these constituents as identical to words. At the same time certain difficulties arise, such as: a) some words, for example first names, will not be in our vocabulary. b) it is not clear how we will handle punctuation – it is obvious that a semicolon carries semantic load, so this information should also be somehow encoded. c) the way we handle words with a common root or versions of the same word in different case or number or gender or inflection etc. is clearly suboptimal, since for all these apparently closely related versions we need completely different representations. The solution preferred by the literature as an answer to the above problems is to match tokens - linguistic elements to sub-words, with an unsupervised learning process on a text sample. Thus, depending on the language we are examining, frequently used sub-words can be identified as linguistic elements. So one word can correspond to one token, but the rule is that we need more than one token for each word. Therefore, the first step in

natural language processing is tokenization, where the input text is broken down into tokens, which generally correspond to subwords. Each token corresponds to a fixed-length vector commonly called a word embedding. The embedding of each token arises as a result of learning. In a second phase, each of the word embeddings is forwarded to the GreekBERT transformer network. We are not interested in any of the use-cases in which GreekBERT has been further refined (e.g. Named Entity Recognition), so we discard the head of the network and keep the last layer of the backbone. This corresponds to a feature vector of dimension 768. Finally, we concatenate the vectors we get for each token separately. We use the average of the vectors as an aggregation function.

**Processing with Neural Network.** The neural network we construct accepts as input the (intermediate) result of GreekBERT, a vector of dimension equal to 768, and is called to produce a vector of dimension equal to 8. This size is related to the types of information we wish to estimate. We describe them in more detail in the next subsection (word categorization). The neural network we use is a simple Multi-Layer Perceptron (MLP), consisting of two hidden layers. The input layer as mentioned is of dimension 768, and the two hidden layers are of dimension 64 and 32 respectively. The final output layer is of dimension 8. We have ReLU activation functions everywhere, except for the output layer where we have a sigmoidal activation to get a probability (0% to 100%) for each category of the outcome. At test time, if a probability is above the 50% threshold, we accept that it corresponds to a positive estimate. Before proceeding, we note that we have optimized the learning process by making extensive use of the data augmentation technique. Initially we had at our disposal a relatively small set of 25 inscriptions which were transcribed by experts. We multiplied the volume of the set by using variations of each word. Specifically, we have considered creating / augmenting with 20 different transcribed inscriptions for each given ground truth inscription, where we replace available words with carefully chosen "variations". By "variations" here we mean one of the following:

- Another spelling for the same word: We apply a random letter change so that the "correct" word appears, but written incorrectly. This was done because, as we noticed in the introductory section of this text, a significant number of the words in the inscriptions of our set are written incorrectly by the scribes - many times by introducing the same word incorrectly in the same inscription. This way of augmentation aims to simulate this data, and at the same time makes the estimation of the network indirectly more "robust" to this kind of "noise".
- Other name in place of main name: We randomly change one name to another. Note that it shouldn't matter whether the name refers to a donor / founder or not, since this information is revealed by the context and never by the name itself. So we change the name keeping its notation constant - if it is a founder it remains a founder, and if it is not a founder it remains as a non-founder. In this step we make use of a list of first names, from which we randomly select the "substitute" name.

– Other date in date slot. Similar to the previous augmentation type, we randomly change the dating words. This in all cases of dating types is relatively simple - we only have to produce a chronology, depending on the original type of chronology (anno mundi, anno domini, indiction) always taking care to stay within the chronological framework of the Byzantine - late Byzantine period.

We replace each word with a variation (when the variations describe apply for the given word) with a 80% chance. We trained our MLP for 150 epochs, using Adam and a learning rate set to $10^{-4}$.

**Word classification.** We adapt our network as a classifier by adding a sigmoid activation. For each word as input, therefore, we get as a result a probability that it belongs to one of the following categories.

1. Word related to the founder of the site.
2. Word related to dating of the site.
3. Word indicating the month of construction.
4. A word that refers to a person.
5. Word indicating the year of construction.
6. Dating is Anno Domini (dating is based on the number of years since the birth of Jesus Christ).
7. Dating is Anno Mundi (dating is based on the number of years since the "creation of the world" in 5509 BC).
8. Indiction dating (dating based on a 15-year repeating cycle).

Categories related to "dating" and the "year" are different taxonomies, as dating may refer to words that could indicate the month or the day for example, or other information describing dating in a periphrastic manner.

Also, note that some of the categories could possibly be formulated as mutually exclusive with respect to others (for example, a dating cannot be an "indiction" dating and Anno Domini at the same time, for the same word). We kept the version above, with all categories as non-mutually exclusive, keeping architecture as simple as possible for our baseline.

### 5.3 Numerical evaluation

Over the total 25 inscriptions with recorded NLP metadata (i.e., as a minimum we require having the full transcription over the text of the inscription, and pointers towards the semantics of each word), The split is done according to the "site id" of each inscription, with the first 16 inscriptions assigned to the training set, and the 9 remaining ones assigned to the test set.

We evaluate the method we described in the previous subsections over three subtasks, considered over each inscription word separately:

1. Is this word related to the founder of the site?
2. Is this word related to the dating of the site?

3. Does this word refer to the month of dating?
4. Does this word refer to a person (not necesarrily one of the founders)?
5. If this word refers to a year, is this type of dating correct?

**Table 5.** Numerical evaluation of the NLP task.

| | Founder Dating | Month word | Person word | Year dating type |
|---|---|---|---|---|
| CC Ratio | 75.5% | 95.7% | 99.7% | 88.5% | 94.4% |

## 6 Concluding Remarks

We have presented a dataset of Byzantine-era Medieval Greek inscriptions that included both text and images, designed to serve as a challenging testbed for Vision and NLP tasks. The scarcity of large related training sets, on either imaging or text data, combined with the historical nature and content of the inscriptions, creates a context in which modern resource-intensive techniques are not straightforward to apply to. We have posed baseline solutions to all three suggested tasks, and when possible we present two different baselines that each fulfill orthogonal requirements; accuracy comes often at a heavy cost in terms of resources.

In the future, we envisage this dataset being updated with new tasks, or "moving the goalpost" to more difficult challenges. For example, a more precise question answering type could be in order, independent of a word classification task. Along those lines, for example an answer in natural language could also be an updated requirement for the NLP task, or considering detection and recognition methods that are both highly accurate and non-resource intensive.

## Acknowledgments

This research has been partially co - financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call : "OPEN INNOVATION IN CULTURE", project *Bessarion* (T6YB$\Pi$ - 00214).

## References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

2. Kaddas, P., Palaiologos, K., Gatos, B., Katsouros, V., Christopoulou, K.: A system for processing and recognition of greek byzantine and post-byzantine documents. In: International Conference on Document Analysis and Recognition. pp. 366–376. Springer (2023)
3. Kaldellis, A.: Byzantine Readings of Ancient Historians: Texts in Translation, with Introductions and Notes. Routledge (2015)
4. Koutsikakis, J., Chalkidis, I., Malakasiotis, P., Androutsopoulos, I.: Greek-BERT: The Greeks visiting sesame street. In: 11th Hellenic Conference on Artificial Intelligence. pp. 110–117 (2020)
5. Miller, J., Krauth, K., Recht, B., Schmidt, L.: The effect of natural distribution shift on question answering models. In: International conference on machine learning. pp. 6905–6916. PMLR (2020)
6. Pavlopoulos, J., Kougia, V., Arias, E.G., Platanou, P., Shabalin, S., Liagkou, K., Papadatos, E., Essler, H., Camps, J.B., Fischer, F.: Challenging error correction in recognised byzantine greek. Research Square Preprints (2024)
7. Retsinas, G., Sfikas, G., Gatos, B., Nikou, C.: Best practices for a handwritten text recognition system. In: International Workshop on Document Analysis Systems. pp. 247–259. Springer (2022)
8. Seuret, M., Marthot-Santaniello, I., White, S.A., Serbaeva Saraogi, O., Agolli, S., Carrière, G., Rodriguez-Salas, D., Christlein, V.: ICDAR 2023 competition on detection and recognition of greek letters on papyri. In: International Conference on Document Analysis and Recognition. pp. 498–507. Springer (2023)
9. Sfikas, G., Giotis, A., Retsinas, G., Nikou, C.: Quaternion generative adversarial networks for inscription detection in byzantine monuments. In: 2nd International Workshop on Pattern Recognition for Cultural Heritage (PatReCH) (2021)
10. Sfikas, G., Ioannidis, D., Tzovaras, D.: Quaternion Harris for multispectral keypoint detection. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 11–15. IEEE (2020)
11. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse R-CNN: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14454–14463 (2021)