École Polytechnique Fédérale de Lausanne

EPFL Master Optional Semester Project

Ukraine's Epigraphy and Cultural Heritage Data Analysis

Yuheng Lu

June 7, 2024

Project Supervisors:
Dr. Hamest Tamrazyan
Dr. Emanuela Boros

Abstract

This project mainly focused on the preservation and usage of Ukrainian Epigraphy and cultural heritage in the format of text(books). In this project, we built a data pipeline for processing targeted at data resources from Digital Laboratory of Ukraine and their storage in database. Then we introduced two language models, SpaCy-based model and BERT-based model, to implement NER tasks on our datasets. After comparisons on both models' performances on test dataset, we chose the SpaCy-based one as our final choice, which can do a good job pn NER classification on 13 different classes of keywords on Ukrainian Epigraphy and cultural heritages.

Contents

Chapter 1

Introduction

Cultural heritage provides a tangible link to the past, offering insights into the historical experiences and achievements of a nation. The preservation of historical sites, monuments, and artifacts ensures that future generations can access and learn from this rich legacy, maintaining a sense of continuity and historical awareness.

This project aims to explore, retrieve, and analyse data from the Digital Laboratory of Ukraine, focusing on a select collection of approximately ten books related to epigraphy and cultural heritage. The primary objective is to gain insights into Ukraine's epigraphic and cultural heritage through detailed data analysis, term extraction, and database management.

1.1 Context

Since 2014, the conflict between Ukraine and Russia has escalated, culminating in the full-scale war that erupted in 2022. This war has had profound impacts on Ukraine's social, economic, political, and cultural landscape. Beyond the immense human toll and the destruction of infrastructure, the conflict poses a significant threat to Ukraine's rich cultural heritage. In this dire context, the preservation of Ukrainian cultural heritage has become an urgent and critical task.Besides, Ukraine's cultural heritage contributes to the cultural diversity of the world. Many Ukrainian cultural sites are recognized as part of the global heritage, including UNESCO World Heritage Sites. Their preservation is not only a national concern but also a global responsibility.

The Digital Laboratory of Ukraine hosts an expansive collection of resources documenting the nation's cultural and historical heritage. Within this extensive repository, a subset of texts pertaining to epigraphy and cultural heritage has much potential to get identified for rigorous analysis. However, these rich Ukrainian cultural materials are in a state of development, and there is no language model designed to analyze and process these materials. By concentrating on these specific resources, the project aims to elucidate the architectural and epigraphic significance of historical sites, particularly ecclesiastical structures, and to develop innovative methodologies for the digitization and visualization of cultural heritage.

1.2 Motivation

Given such abundant resources in format of books and texts, we want to make best use of them. However, there are some obstacles when we deal with the data analysis on these text resources. Firstly, most of these data resources are still in a raw state, i.e. in .pdf format. It contains a lot

of useless information and is also hard to process. Besides, it is impossible to find a developed language model that is specified for epigraphy keywords detection in Ukrainian. Therefore, for implementing researches on these resources, firstly there is a great demand for a full pipeline for dealing with raw dataset into a uniform format, which filters out all the unnecessary information and preserves only needed texts for epigraphy and cultural heritage. Then we need to find a suitable format to store them into database. Finally, a new natural language model that is trained specifically for Ukraine's cultural heritages, which mainly focuses on NER task for epigraphy keywords detection and their classification. Therefore, this project will contribute to the understanding of Ukraine's rich cultural heritage. It will provide valuable digital resources for future academic and cultural research in this field.

1.3 Goals

This project aims to deepen the knowledge of the architectural and epigraphic significance of the church, explore innovative techniques for digitizing and visualizing cultural heritage, and contribute to the preservation and accessibility of Armenian inscriptions in Nagorno-Karabakh.

To be specific, the main goals for this project can be divided into four parts as follows:

- **Data Retrieval:** Collect and aggregate data from the Digital Laboratory of Ukraine, specifically targeting books and resources about epigraphy and cultural heritage.
- **Data Cleaning and Formatting:** Implement data preprocessing techniques to ensure data quality, including removing irrelevant or corrupt data, handling missing values, and standardizing formats.
- **Database Setup:** Design and implement a database to store and manage retrieved data efficiently, allowing easy access and manipulation for analysis.
- **Term Extraction and Analysis:** Employ natural language processing (NLP) techniques to extract key terms, concepts, and thematic elements from the texts to understand predominant themes and patterns in Ukrainian epigraphy and cultural heritage.

Chapter 2

Data pipeline

In order to implement this project, a good data pipeline should be introduced so that the data quality can be good for training. The whole data pipeline for processing the raw PDF data includes data retrieval, data cleaning & formatting and database setup. In this chapter, the methods and processes of these three tasks will be illustrated.

2.1 Data Retrieval

Data retrieval mainly focuses on collecting and aggregating data from the Digital Laboratory of Ukraine. Despite getting the PDF files from Digital Laboratory of Ukraine, we still need to do annotations on the collected data.

We use Sketch Engine(Wikipedia 2024a) to help us with keyword extraction in constructing training dataset. Sketch Engine is a powerful text analysis and corpus management tool widely used in linguistic research, lexicography, language teaching, and natural language processing. It allows users to search, analyse, and manage large-scale language data. Besides, it automatically generates collocation patterns and usage examples for specific words, helping users understand word usage in different contexts. It can also be used to extract significant keywords from texts, helping users identify the main themes and content. Helped by Sketch Engine, Hamest Tamrazyan manually create a list of keywords in Ukrainian Epigraphy and cultural heritage, which helps us with the preparation of annotations of training datasets.

2.2 Data Cleaning & Formatting

After collecting the PDF files, we use transforming tools to convert them into text files so that these books can be processed through coding.

We mainly use regex to deal with these books. The books have a general structure, see figure 2.1. A book has information about publication information, introduction, chapters, conclusion and afterwords, appendix. Each chapter has many sections and each section has many graffiti. Each graffiti has at most 4 elements: location, description, dating and category. This clear structure allows us to process data cleaning and data formatting by designing standard progress:

- **For useless pages:** In one book, there are information about publication and introduction to this book. Besides, there are also conclusion, afterwords and appendix. All these pages are not needed and we should delete them.

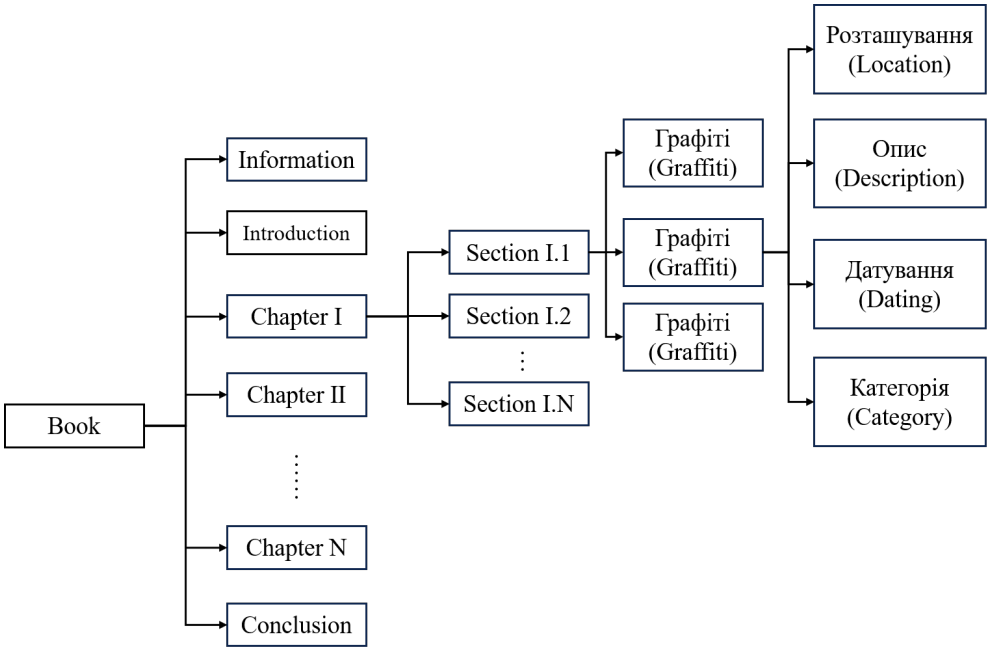

Figure 2.1
Book Structure

- **For figure comment:** After converting to format of text file, the figures in original file are deleted but their comments are kept. Therefore, these comments should be deleted as well.
- **For text segment:** After conversion, the text files have many "↓" s that seperate a whole paragraph into different parts. It is a must to restore the original paragraph by deleting these "↓"s by regex matching and deletion.
- **For necessary components:** Use regex to match the necessary part of the text and and preserve all useful information.

After processing the original text file, the remaining part is structured with only demanded components, see Figure 2.2.

2.3 Database Setup

Although all the information in the text files are cleaned and formatted, it is still hard to retrieve key information efficiently in a text file since it is unstructured and does not contain any index. In order to store the text information in a retrieving-friendly way, two ways of data storage are designed in database setup.

Two formats for database setup is taken into consideration:

- **Use NoSQL format:** Non-relational databases employ more flexible data models, such as document, key-value, column-family, and graph databases, allowing for the storage of complex and diverse data structures. Document-oriented databases (e.g., MongoDB) store data in JSON or BSON format, which is adept at handling nested and semi-structured data. Key-value stores are optimized for quick lookups and simple transactions. Since the remaining text are clearly structured in a NoSQL format(key-value pair), it is easy to store this information in a JSON file.

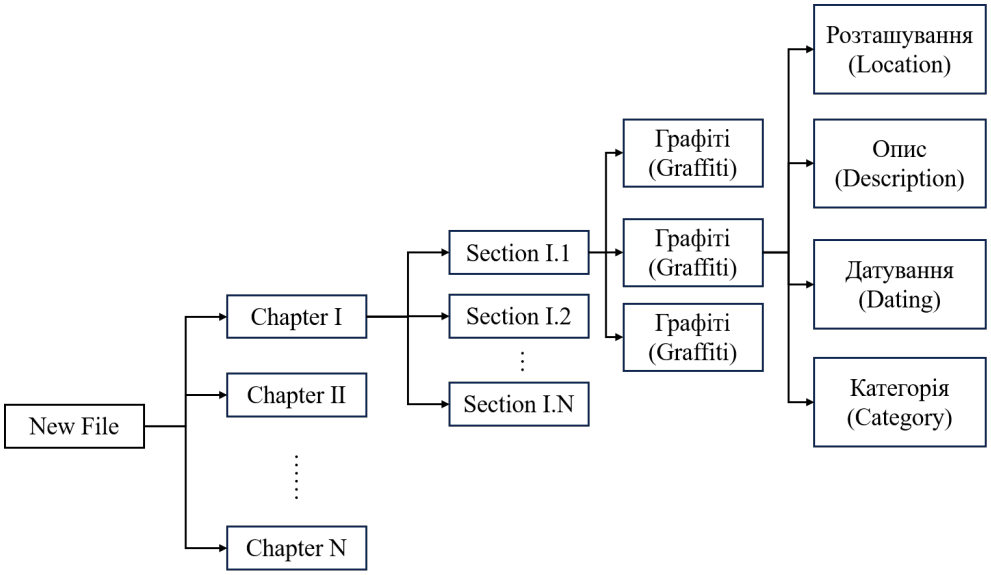

Figure 2.2
New Text File Structure

- **Use Relational format:** The relational database model stores data in tables, which are organized into rows and columns. Each table represents an entity type, with each row corresponding to a specific instance of that entity, and each column representing an attribute of the entity. By defining primary keys and foreign keys, relational databases efficiently manage data integrity and support complex querying operations. When implementing the NER task, it is convenient to store the IOB information of tokens in sentences by relational format file like CSV. This will be illustrated in chapter 3.2.

By employing these two different data storage formats, it is possible to select an appropriate database format based on specific requirements, thereby enhancing the efficiency and accuracy of information retrieval.

Chapter 3

Keywords Extraction by NER

In this chapter, we explore the application of advanced natural language processing (NLP) techniques for the Named Entity Recognition (NER)(Wikipedia 2024b) task, specifically tailored for Ukrainian epigraphy and cultural heritage texts. This task involves identifying and classifying entitie of cultural artifacts within the textual data. Given the complexity and historical significance of Ukrainian epigraphy and cultural heritage, precise and efficient NER is crucial for digitizing, cataloging, and analyzing these texts.

We employ two state-of-the-art NLP frameworks, BERT (Bidirectional Encoder Representations from Transformers) and SpaCy, to perform NER. BERT(Devlin et al. 2019), a deep learning model developed by Google, has revolutionized NLP with its transformer-based architecture, enabling it to capture contextual relationships in text more effectively than traditional models. SpaCy(Wikipedia 2024c), a robust and efficient NLP library, provides an accessible and high-performance framework for processing and analyzing large volumes of text. In this chapter, we will illustrate how we implement these two techniques to get a language model specified for our task.

3.1 SpaCy-based model

SpaCy(Wikipedia 2024c) is a powerful and versatile open-source library designed for advanced Natural Language Processing (NLP) in Python. It is widely recognized for its efficiency, speed, and ease of use, making it a preferred choice for developers and researchers working on NLP tasks. SpaCy provides a broad range of features and functionalities, enabling the processing and analysis of large volumes of text with remarkable accuracy and performance.

We select SpaCy for the following reasons: First, SpaCy is built for production use, designed to handle real-world tasks and large datasets efficiently. Its architecture is optimized for speed and memory usage, making it suitable for applications that require high performance; Second, SpaCy offers pre-trained statistical models for multiple languages, including English, Spanish, French, German, and of course Ukrainian. These models enable immediate out-of-the-box functionality for various NLP tasks such as tokenization, part-of-speech (POS) tagging, named entity recognition (NER), and dependency parsing. Last but not least, SpaCy's NER component is particularly robust, capable of identifying and classifying named entities like people, organizations, dates, and locations within a text. Users can also train custom NER models to recognize domain-specific entities. Given so many advantages, it is a good option for implementing our language model based on SpaCy that supports Ukrainian and train this blank model based on

our own datasets.

Figure 3.1 illustrates the process of training a SpaCy model.

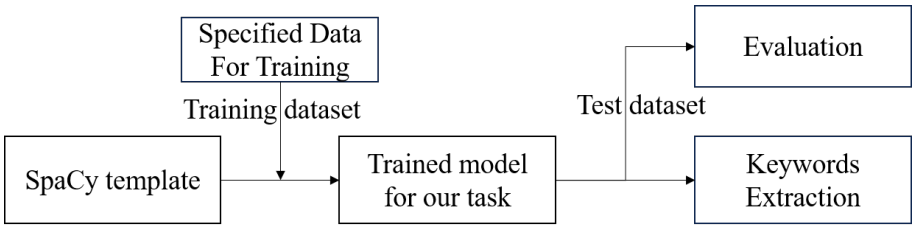

Figure 3.1
SpaCy process

Following such process, we start to deal with our datasets to fit in with the required format to train the SpaCy model. Then we use the training dataset to train the model, use the test dataset and three metrics: Accuracy, Recall and F1-score to evaluate the model's performance. Finally, we will apply this model to detect new keywords in new texts.

3.1.1 IOB format

In order to do keyword extraction in our task, we should label the keywords in our training datasets with IOB formatting. The Inside-Outside-Beginning (IOB) format(Wikipedia 2024d) is a widely used annotation scheme in Natural Language Processing (NLP) for representing the boundaries of entities in text sequences. It plays a crucial role in the field of NLP, providing a clear and consistent method for annotating text data for entity recognition tasks. Its structure allows for efficient training and evaluation of Named Entity Recognition (NER) models, contributing significantly to advancements in the automatic extraction and classification of entities from text. By adopting the IOB format, researchers and practitioners can ensure high-quality, interoperable annotations that facilitate robust NLP applications.

In the IOB scheme, each token in a text is assigned a label that indicates whether it is inside, outside, or at the beginning of a named entity. The labels follow a specific convention:

- **B-type:** Indicates that the token is at the beginning of an entity of type.
- **I-type:**Indicates that the token is inside an entity of type.
- **O:**Indicates that the token is outside an entity.

There are 13 types of keywords in total: inscription', 'dating_criteria', 'execution_technique', 'monument_subtype', 'material', 'epigraphic_shorthand', 'object_type','symbol', 'inscription_type', 'preservation_state', 'monument', 'decoration', 'other'. Here is an example of our labeled sentence using IOB format in our dataset, see figure 3.2.

In this example:

- **'молитовне'** is labeled 'B-inscription_type' to indicate it is the beginning of a inscription_type entity.
- **'звернення'** is labeled 'B-inscription_type' to indicate it is inside an inscription_type entity.
- **Other tokens** are labeled 'O' to indicate they do not belong to any entity type in our dataset.

На	O
фресці	O
виконане	O
молитовне	B-inscription_type
звернення	I-inscription_type
:	O
Помени	O
Марка	O
.	O

Figure 3.2
IOB annotation example

3.1.2 Dataset Preparation

Training a SpaCy model, particularly for tasks such as Named Entity Recognition (NER), requires a well-structured and annotated dataset. The quality and format of this training data are crucial for the model's performance and accuracy. This section delves into the required format for training data in SpaCy, highlighting its structured nature and the specific annotations needed.

The annotated training data for NER in SpaCy follows a specific structure, typically in the form of a list of tuples. Each tuple consists of two elements: the text itself and a dictionary containing annotations for named entities within that text.

The structure of annotated data consists of two parts. The first part is the sentence text: each tuple starts with the text that contains named entities to be recognized. This text is usually a sentence, but sometimes can be a paragraph, or any other unit of text. In our project, we select this text to be a sentence. The second part is annotations dictionary containing annotations for the named entities in the text. This dictionary has a key named "entities", which maps to a list of tuples. This name cannot be changed into other names. Each tuple within the "entities" list represents an annotated named entity. It contains three elements:

- **Start position:** the character position where the entity starts within the text.
- **End position:** the character position where the entity ends within the text.
- **Label:** the label or type of the named entity.

Figure 3.3 is an example for our datasets that are transformed into the NER task required format.

```
('На фресці прокреслений чотириконечний хрест , основа щогли якого спи- рається на
відкриту донизу одноступінчасту Голгофу .',
  {'entities': [(10, 22, 'execution_technique'),
    (23, 45, 'decoration'),
    (46, 64, 'object_type')]})
```

Figure 3.3
NER format dataset

In this example, there are three keywords in one sentence. All their start positions, end positions

and keyword type are included in 'entities'.

3.1.3 Train the model

The training process for this NER model using SpaCy is designed as follows: It iterates through 25 epochs, each time shuffling the training data and dividing it into mini-batches. The mini-batches are dynamically adjusted in size, gradually increasing from 4 to 32. Within each mini-batch, the NLP model is updated using annotated text examples, with a dropout rate of 0.5 applied to prevent overfitting. Losses incurred during each epoch are monitored to track the model's training progress and convergence. This approach ensures efficient and effective training of the NER model, ultimately enhancing its ability to accurately identify named entities within text.

The training loss trend plot can be seen in Figure 3.4.

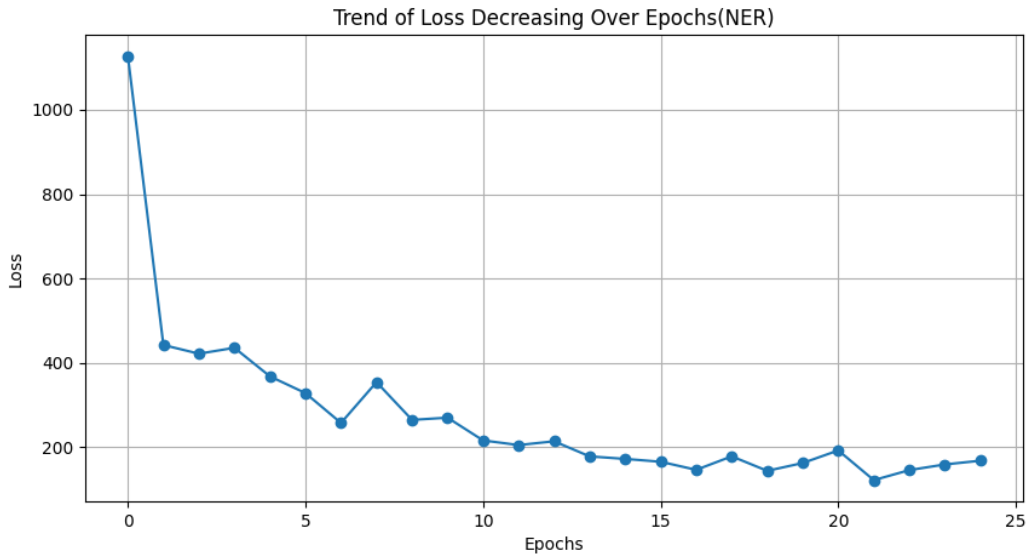

Figure 3.4
Training loss trend (NER)

3.1.4 Test the model

When testing our NER model, a summary similar to a classification report is typically generated. This summary includes the precision, recall, and F1-score for each named entity category, as well as metrics like micro-average and macro-average. Here's a brief explanation of these metrics:

- **Precision:** Precision refers to the proportion of samples predicted as a particular category by the model that actually belong to that category. In NER tasks, precision indicates the model's ability to correctly identify named entities. The formula for precision is given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

  where:

  – TP (True Positives) is the number of correctly predicted positive samples.

  – FP (False Positives) is the number of incorrectly predicted positive samples.

- **Recall:** Recall is the proportion of samples actually belonging to a particular category that the model successfully predicts as that category. In NER tasks, recall indicates whether the model can capture all named entities belonging to that category. The formula for recall is given by:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  where:

  – TP (True Positives) is the number of correctly predicted positive samples.

  – FN (False Negatives) is the number of incorrectly predicted negative samples.

- **F1-score:** The F1-score is the harmonic mean of precision and recall. It provides a balanced measure between precision and recall, taking both false positives and false negatives into account. In NER tasks, the F1-score reflects the overall performance of the model in identifying named entities. The formula for F1-score is given by:

$$F1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

  where:

  – Precision is the ratio of correctly predicted positive observations to the total predicted positives.

  – Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

- **Micro-average Macro-average and weighted-average:** Micro-average calculates the metrics globally by considering all samples collectively, while macro-average calculates the metrics independently for each class and then takes the average. Micro-average is useful when class imbalance is present, while macro-average provides insights into the model's performance on individual classes.In the context of classification model evaluation, weighted average is a technique used to account for class imbalances by considering the contribution of each class according to its proportion in the dataset. This method provides a more accurate reflection of the model's performance across different classes, especially when some classes have significantly more samples than others.

  The formulas for micro-average precision, recall, and F1-score are given by:

$$\text{Micro-average Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FP}}$$

$$\text{Micro-average Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}}$$

$$\text{Micro-average F1-score} = 2 \cdot \frac{\text{Micro-average Precision} \cdot \text{Micro-average Recall}}{\text{Micro-average Precision} + \text{Micro-average Recall}}$$

  The formulas for macro-average precision, recall, and F1-score are given by:

$$\text{Macro-average Precision} = \frac{1}{N} \sum_{i=1}^{N} \text{Precision}_i$$

$$\text{Macro-average Recall} = \frac{1}{N} \sum_{i=1}^{N} \text{Recall}_i$$

$$\text{Macro-average F1-score} = \frac{1}{N}\sum_{i=1}^{N}\text{F1-score}_i$$

where N is the number of classes.

The formulas for weighted average precision, recall, and F1-score are given by:

$$\text{Weighted Precision} = \frac{\sum_{i=1}^{N}(\text{Precision}_i \cdot \text{Support}_i)}{\sum_{i=1}^{N}\text{Support}_i}$$

$$\text{Weighted Recall} = \frac{\sum_{i=1}^{N}(\text{Recall}_i \cdot \text{Support}_i)}{\sum_{i=1}^{N}\text{Support}_i}$$

$$\text{Weighted F1-score} = \frac{\sum_{i=1}^{N}(\text{F1-score}_i \cdot \text{Support}_i)}{\sum_{i=1}^{N}\text{Support}_i}$$

where:

- Precision_i is the precision for class i.
- Recall_i is the recall for class i.
- F1-score_i is the F1-score for class i.
- Support_i is the number of true instances for class i.
- N is the number of classes.

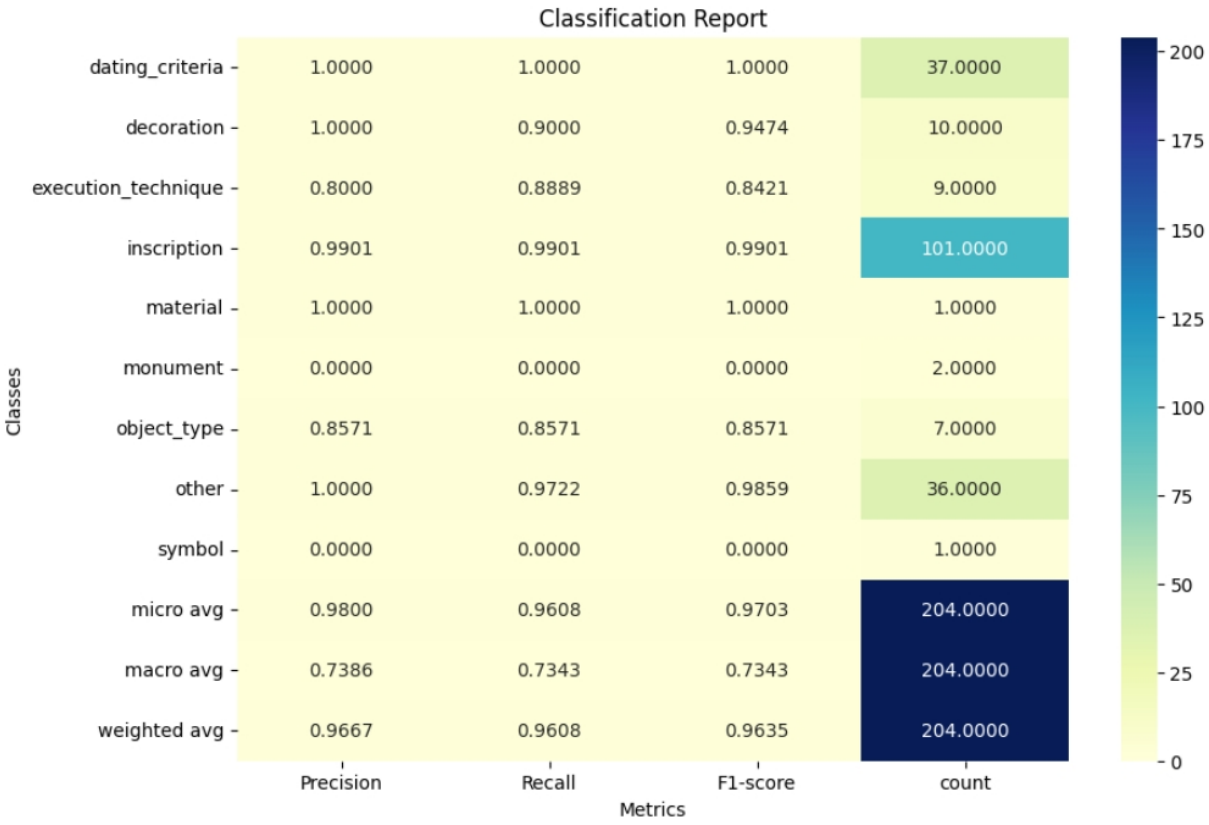

Figure 3.5
NER Classification report

In summary, the classification report summarizes the performance of the NER model across different named entity categories, providing insights into its precision, recall, and overall effectiveness in identifying named entities in the text. In the test dataset, the NER model demonstrates high performance overall, with particularly strong results for classes like ”dating_criteria,” ”inscription,” and ”other.” However, the model shows weaknesses in identifying less frequent classes such as ”monument” and ”symbol” where all metrics are zero. These results suggest that while the model is reliable for the majority of entity types, it may require further training and potentially more data for the underperforming classes to improve its robustness across all entity types.

3.1.5 Detect new keywords

After successfully training a Named Entity Recognition (NER) model using SpaCy, the next step is to leverage this model to detect entities in new, unseen text. This process involves several steps that transform raw text data into structured information through the pre-trained model. Below is an explanation of the principles behind this process and a detailed workflow description.

Figure 3.6 illustrates count of new keywords detected group by their type.

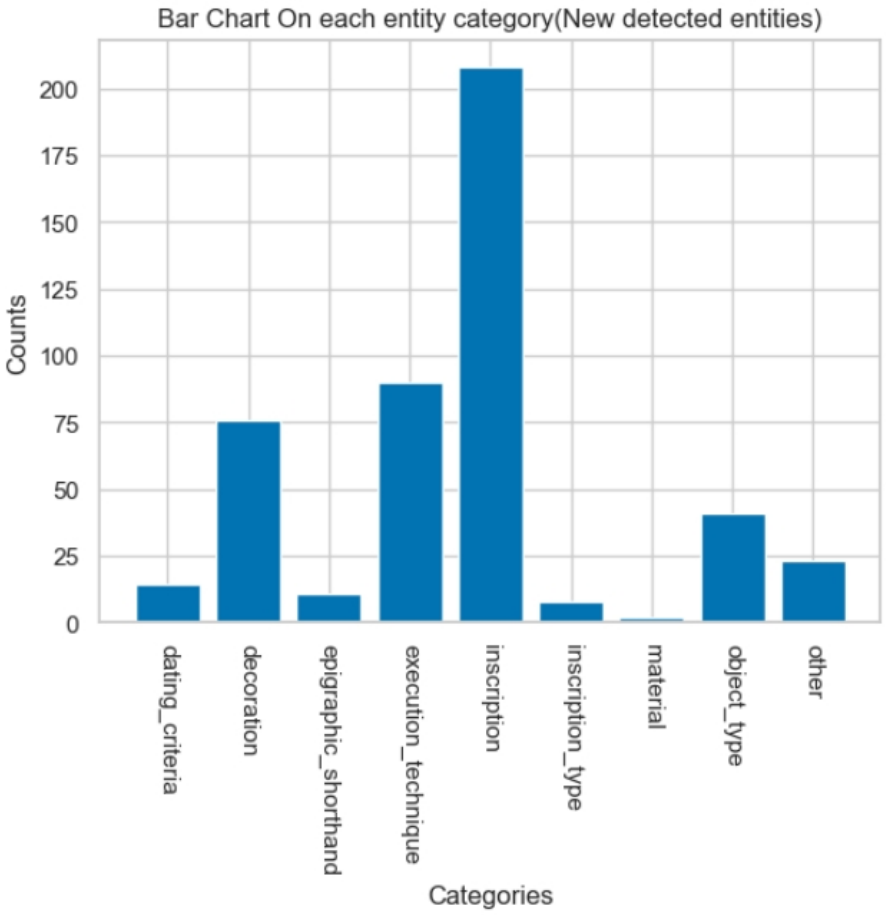

Figure 3.6
New Detected keywords in groups

3.2 BERT-based model

BERT(Devlin et al. 2019), which stands for Bidirectional Encoder Representations from Transformers, is a groundbreaking model introduced by researchers at Google AI Language in 2018. It represents a significant advancement in natural language processing (NLP) by providing deep bidirectional context understanding, which allows the model to capture the meaning of words based on their surrounding context more effectively than previous models. BERT is based on the transformer architecture, which uses self-attention mechanisms to weigh the importance of different words in a sentence. This helps in capturing the nuanced relationships between words in a text.

BERT involves two stages: pre-training and fine-tuning.During pretraining, BERT learns general language understanding from a large corpus of text. In the fine-tuning stage, BERT is specialized for specific tasks (like NER) by training on a smaller, task-specific dataset. In our project, we also import a pretrained BERT model that supports Ukrainian and do fine-tuning with the same dataset we used for training SpaCy.

3.2.1 BERT-NER

BERT-NER leverages BERT's powerful contextual embeddings to enhance the accuracy and performance of NER systems. Therefore, it requires less task-specific training data to achieve high performance compared to traditional NER models. Besides, it benefits from BERT's bidirectional context understanding, leading to more accurate entity recognition even in complex sentences. BERT-NER works under the following principles:

- **Tokenization:** The input text is first tokenized into subwords using BERT's WordPiece tokenizer. This handles out-of-vocabulary words more effectively by breaking them into known subword units.
- **Embedding:** The tokenized input is then converted into embeddings using BERT's pretrained model. Each token is represented as a dense vector that captures its contextual meaning.
- **Feature Extraction:** The embeddings are passed through BERT's transformer layers to capture deeper contextual relationships. This results in contextualized embeddings for each token, enriched with information from surrounding tokens.
- **Classification Layer:** On top of the BERT model, a classification layer is added specifically for NER. This layer assigns entity labels to each token based on its contextual embeddings.
- **Fine-tuning** BERT-NER is fine-tuned on a labeled NER dataset. The pre-trained BERT model is adapted to the NER task by training it on annotated text, allowing it to learn to identify and classify entities accurately.

3.2.2 Dataset Preparation

BERT-NER dataset format is crucial for training named entity recognition (NER) models using BERT-based architectures. This format ensures that the training data is structured in a way that allows the model to effectively learn to recognize and classify entities in text.

The dataset consists of tokenized text sequences, where each sequence represents a sentence or a segment of text. Tokenization is performed using BERT's WordPiece tokenizer or similar methods, which split the text into subword units to handle out-of-vocabulary words effectively.

Each token in the tokenized text sequences is associated with a label that indicates its entity type or its relation to an entity. These labels are typically encoded using a numerical scheme, such as IOB (Inside, Outside, Beginning) encoding.

The BERT-NER dataset is structured as a collection of samples, with each sample containing two main components: the tokenized text sequence and the corresponding token-level labels. These samples are typically stored in a JSON or similar format for easy parsing and processing. In our dataset, we use embedding to convert all the ner_tags in numbers. Below is figure 3.7, an example of the BERT-NER dataset format from our dataset:

{'tokens': ['На','фресці', 'прокреслений', 'чотириконечний', 'хрест', ',',
'основа', 'щогли', 'якого', 'спи-', 'рається', 'на', 'відкриту', 'донизу',
'одноступінчасту', 'Толгофу', '.'],
'ner_tags': [1, 1, 10, 16, 9, 1, 12, 20, 1, 1, 1, 1, 1, 1, 1, 1, 1]}

Figure 3.7
BERT-NER dataset example

And the embedding pairs are shown in figure 3.8.

{'I-monument': 0, 'O': 1, 'B-symbol': 2, 'B-inscripiton type': 3,
'I-preservation_state': 4, 'B-monument': 5, 'B-dating_criteria': 6,
'I-material': 7, 'B-other': 8, 'I-decoration': 9,
'B-execution_technique': 10, 'I-other': 11, 'B-object_type': 12,
'B-inscription': 13, ,B-inscription_type': 14, ,I-inscription_type': 15,
,B-decoration': 16, ,B-material': 17, 'B-preservation_state': 18,
'I-inscripiton type': 19, 'I-object_type': 20, 'B-epigraphic_shorthand': 21}

Figure 3.8
BERT-NER embedding pairs

3.2.3 Train the model

We train this model and the training loss and validation loss can be seen in figure 3.9. Since the overfitting appears at epoch 6, we choose epoch as 5. The learning rate is 0.00002, the batch size is 4 and the weight decay is selected as 0.01. It can be seen that both two losses are decreasing to a low value.

3.2.4 Test the model

Testing result can be seen in figure 3.10. Compared with the result of SpaCy-based model, bert-based model's performance is apparently worse. Therefore, we preferred the SpaCy-based model for our final choice. And therefore, we do not use bert-based model to do new keywords' detection.

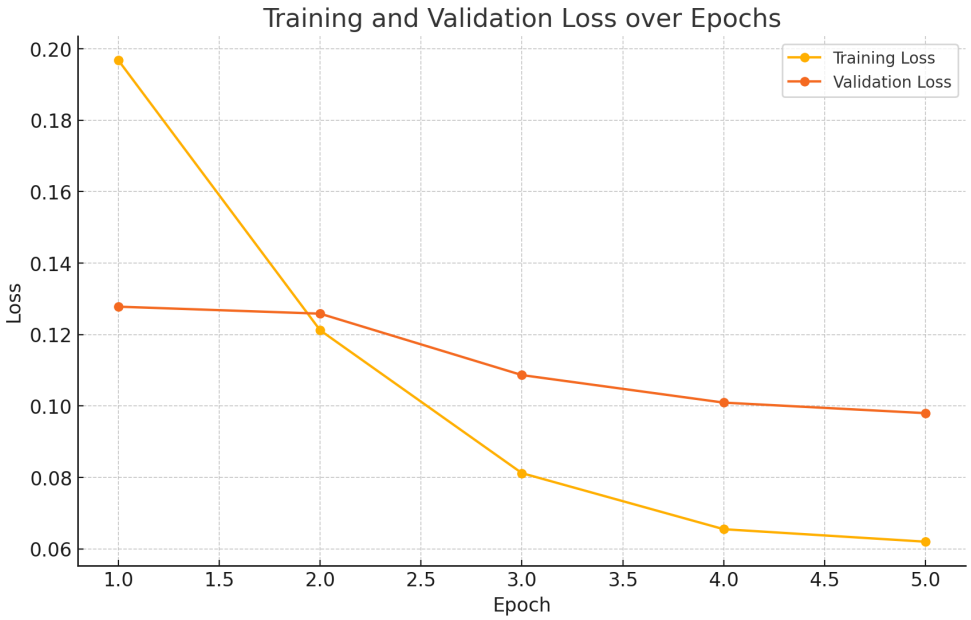

Figure 3.9
BERT-Training loss and validation loss

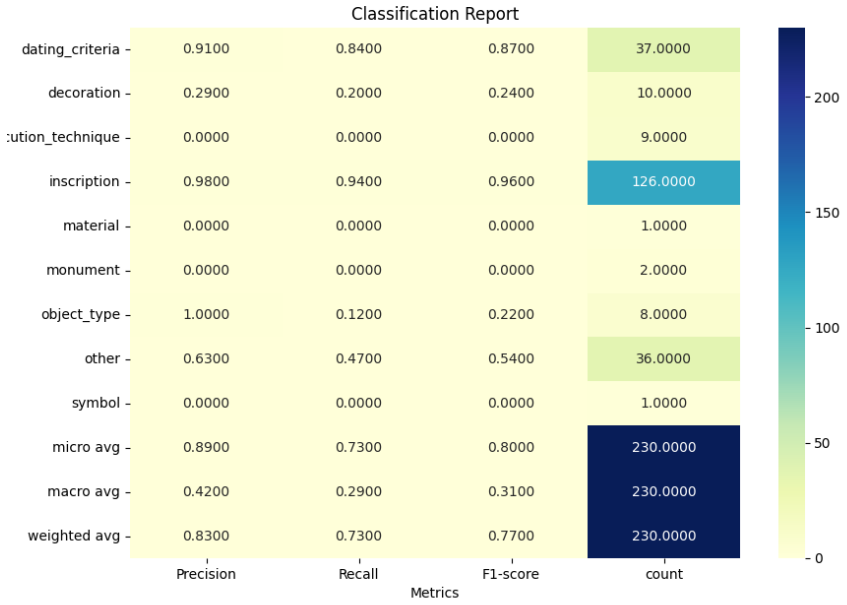

Figure 3.10
BERT-classification report

Chapter 4

Relation with SKOS

4.1 SKOS

SKOS, which stands for Simple Knowledge Organization System, is a W3C recommendation designed to provide a common model for expressing knowledge organization systems such as thesauri, classification schemes, and taxonomies. SKOS provides a set of concepts and relationships that enable the creation, publication, and management of vocabularies and structured knowledge representations on the web. It allows for the formal representation of concepts, their labels, and the relationships between them, facilitating interoperability and knowledge discovery across different systems and domains.

SKOS consists of several key components:

1. **Concepts**: These are the basic units of knowledge represented in SKOS. Concepts represent the subjects or ideas that are being described or classified.
2. **Labels**: Concepts are associated with labels, which are human-readable terms used to identify and describe them. Labels can include preferred labels, alternative labels, and hidden labels.
3. **Hierarchical Relationships**: SKOS supports hierarchical relationships between concepts, allowing them to be organized in a hierarchical structure such as a taxonomy or a thesaurus.
4. **Associative Relationships**: In addition to hierarchical relationships, SKOS allows for the specification of associative relationships between concepts, capturing non-hierarchical relationships such as related, broader, narrower, and related concepts.
5. **Mappings**: SKOS provides mechanisms for expressing mappings between concepts in different knowledge organization systems, enabling interoperability and integration across diverse vocabularies.

4.2 SKOS's relation with our project

The use of our model within the context of SKOS can greatly enhance the organization, discovery, and dissemination of knowledge related to Ukrainian cultural heritage.

1. **Concept Extraction**: NER models trained specifically for Ukrainian cultural heritage can automatically extract named entities from text data. These extracted entities can be

mapped to concepts in the SKOS vocabulary, enriching the vocabulary with specific terms and entities relevant to Ukrainian epigraphy and cultural heritage.

2. **Semantic Annotation**: The extracted named entities from our model can be semantically annotated with SKOS concepts, associating them with relevant concepts in the SKOS vocabulary. This allows for the integration of named entity information into the broader knowledge organization framework provided by SKOS, enabling users to navigate and explore Ukrainian cultural heritage information within the context of existing knowledge structures.
3. **Interoperability and Integration**: By mapping named entities to SKOS concepts, NER models facilitate interoperability and integration with other knowledge organization systems and vocabularies that adhere to the SKOS standard. This enables seamless integration of Ukrainian cultural heritage information with other cultural heritage resources and knowledge repositories on the web.
4. **Enrichment of SKOS Vocabulary**: The use of NER models helps to enrich the SKOS vocabulary with new terms and entities specific to Ukrainian cultural heritage, thereby enhancing the richness and diversity of the vocabulary and making it more comprehensive and inclusive.
5. **Development of Standardized Vocabulary**: The use of NER models can help retrieve relevant terms and compile a harmonized vocabulary specifically for Ukrainian Epigraphy. These terms can then be organized into a standardized vocabulary based on the SKOS.

In conclusion, the integration of NER models for Ukrainian cultural heritage with SKOS provides a powerful framework for organizing, managing, and sharing knowledge related to Ukrainian cultural heritage in a structured and interoperable manner. By leveraging SKOS as a standard representation model, NER models can effectively contribute to the preservation and promotion of Ukrainian cultural heritage on the web.

Chapter 5

Conclusion

5.1 Conclusion

In this report, we first designed a data pipeline for processing the book-format materials from the Digital Laboratory of Ukraine and kept the useful information in a database. Then we explored the application of advanced natural language processing (NLP) techniques for the Named Entity Recognition (NER) task, specifically tailored for Ukrainian epigraphy and cultural heritage texts. Given the complexity and historical significance of Ukrainian epigraphy and cultural heritage, precise and efficient NER is crucial for digitizing, cataloguing, and analysing these texts.

We employed two state-of-the-art NLP frameworks, BERT (Bidirectional Encoder Representations from Transformers) and SpaCy, to perform NER. BERT, a deep learning model developed by Google, has revolutionized NLP with its transformer-based architecture, enabling it to capture contextual relationships in text more effectively than traditional models. SpaCy, a robust and efficient NLP library, provides an accessible and high-performance framework for processing and analysing large volumes of text. Comparing two models' performance, the SpaCy-based model works better. Therefore, we select the Spacy-based model as our final model for our NER task.

5.2 Future Work Preview

1. **Expansion of Entity Types**: Currently, the NER model may focus on recognizing specific entity types, such as a small number of types in Ukrainian epigraphy and cultural heritage. In order to improve the model's performance, we only set 13 different types. All the less significant keywords are labelled 'other'. Future development could involve expanding the range of entity types to include additional categories relevant to Ukrainian cultural heritage.
2. **Integration with Domain-Specific Knowledge Graphs**: Integrating the NER model with domain-specific knowledge graphs or ontologies related to Ukrainian cultural heritage could enhance its understanding of semantic relationships and contextual information. This integration could enable more sophisticated entity recognition and entity linking capabilities, leading to richer and more accurate annotations.
3. **Multimodal NER**: Incorporating multimodal data sources such as images, videos, or audio recordings alongside text data could enrich the NER model's understanding of Ukrainian cultural heritage. Multimodal NER techniques could be explored to jointly

analyse and extract entities from diverse data modalities, providing a more comprehensive understanding of cultural artifacts and events.

4. **NER model with for automatic translation**: Extending the NER model to support automatic translate will allows for different researchers to better doing research on Ukrainian cultural heritage.

5. **Interactive Knowledge Exploration Tools**: Developing interactive visualization and exploration tools that leverage the NER model's output could empower users to interactively explore and navigate Ukrainian cultural heritage data. These tools could provide intuitive interfaces for querying, browsing, and visualizing entities, enabling users to discover connections and insights within the cultural heritage corpus.

6. **Evaluation and Benchmarking**: Continuous evaluation and benchmarking of the NER model against diverse datasets and evaluation metrics are essential for assessing its performance and identifying areas for improvement. Establishing standardized benchmarks and evaluation protocols specific to Ukrainian cultural heritage NER tasks could guide future development efforts and facilitate comparison with existing models.

7. **Community Engagement and Collaboration**: It is still very hard to find enough pretrained Ukrainian language models in open-source community, which adds more difficulty to our research. Engaging with domain experts, cultural heritage institutions, and local communities in Ukraine can provide valuable insights and feedback for refining the NER model. Collaborative initiatives could involve co-designing annotation guidelines, collecting annotated datasets, and co-developing applications that leverage the NER model to support cultural heritage research and preservation efforts.

By pursuing these future developments and opportunities, the trained NER model for Ukrainian cultural heritage can continue to evolve and contribute to the exploration, documentation, and promotion of Ukraine's rich cultural heritage for generations to come.

Bibliography

Wikipedia (2024a). *Sketch Engine — Wikipedia, The Free Encyclopedia*. [Online; accessed 5-June-2024]. URL: https://en.wikipedia.org/wiki/Sketch_Engine.

– (2024b). *Named-entity recognition — Wikipedia, The Free Encyclopedia*. [Online; accessed 5-June-2024]. URL: https://en.wikipedia.org/wiki/Named-entity_recognition.

Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Wikipedia (2024c). *SpaCy*. Wikimedia Foundation. URL: https://en.wikipedia.org/wiki/SpaCy (visited on June 5, 2024).

– (2024d). *Inside–outside–beginning (tagging)*. Wikimedia Foundation. URL: https://en.wikipedia.org/wiki/Inside%E2%80%93outside%E2%80%93beginning_(tagging) (visited on June 5, 2024).