applied sciences

MDPI

*Article*

# An Approach to a Linked Corpus Creation for a Literary Heritage Based on the Extraction of Entities from Texts

**Kenan Kassab and Nikolay Teslya ***

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 14th line, 39, 199178 St. Petersburg, Russia; Kassab.K@iias.spb.su
* Correspondence: teslya@iias.spb.su

**Abstract:** Working with the literary heritage of writers requires the studying of a large amount of materials. Finding them can take a considerable amount of time even when using search engines. The solution to this problem is to create a linked corpus of literary heritage. Texts in such a corpus will be united by common entities, which will make it possible to select texts not only by the occurrence of certain phrases in a query but also by common entities. To solve this problem, we propose the use of a Named Entity Recognition model trained on examples from a corpus of texts and a database structure for storing connections between texts. We propose to automate the process of creating a dataset for training a BERT-based NER model. Due to the specifics of the subject area, methods, techniques, and strategies are proposed to increase the accuracy of the model trained with a small set of examples. As a result, we created a dataset and a model trained on it which showed high accuracy in recognizing entities in the text (the average F1-score for all entity types is 0.8952). The database structure provides for the storage of unique entities and their relationships with texts and a selection of texts based on the entities. The method was tested for a corpus of texts from the literary heritage of Alexander Sergeevich Pushkin, which is also a difficult task due to the specifics of the Russian language.

**Keywords:** natural language processing; Named Entity Recognition; Bidirectional Encoder Representations from Transformers (BERT); multilingual models; text processing



**Citation:** Kassab, K.; Teslya, N. An Approach to a Linked Corpus Creation for a Literary Heritage Based on the Extraction of Entities from Texts. *Appl. Sci.* **2024**, *14*, 585. https://doi.org/10.3390/app14020585

Academic Editors: José Ramón Méndez Reboredo and David Ruano-Ordás

Received: 24 November 2023
Revised: 3 January 2024
Accepted: 4 January 2024
Published: 9 January 2024



## 1. Introduction

Each writer's heritage contains a great amount of various types of works, from the writer's works, letters, and notes to the scientific papers, critics, memoirs, encyclopedia articles, etc. Conducting any kind of research based on heritage materials without the use of technical means is a very difficult task. The difficulty lies in both the significant volume of materials and the lack of obvious connections between them that leads researchers to spend a great amount of time to find appropriate sources and material through all of the writer's heritage. For example, dozens of scientific works, notes, and critiques can be written about any of the works, both during the writer's lifetime and after his death. A certain person can be a friend of the writer and a prototype of one of the heroes of the work or the hero himself. These types of relationships can be formed using natural language processing (NLP) approaches and stored in a database. Such a structure can be considered as a corpus of linked texts and used in digital humanities research. This corpus could provide a deeper understanding of the various literary works, allowing for the robust visualization of their relationships and enabling the search for more comprehensive and deep information within the texts. All these materials are connected by their mention of the writer's literary works, including some persons who interacted with the author, places where the author was, dates, and organizations connected with the author and the literary work (Figure 1).

The research presented in this work is aimed to create a linked corpus over the literary heritage of the great Russian poet Alexander Sergeevich Pushkin. Such a corpus would contain links between heritage parts based on the common entities between them. To extract

these entities, we propose to use one of the Natural Language Processing approaches related to the search for named entities—Named Entity Recognition (NER)—and build a database which will store entities itself, including links from entities to texts in the corpus that, in general, will provide links between texts. Since the main language of the A.S. Pushkin heritage is Russian, we additionally address some language-specific nuances, but the whole approach can be adapted to any language and any writer. When considering linguistic variety, given that languages such as the Russian language have particular nuances, creating and implementing customized NER systems becomes even more important. This paper also focuses on creating and deploying a unique NER system designed for the processing of the writer's literary heritage.

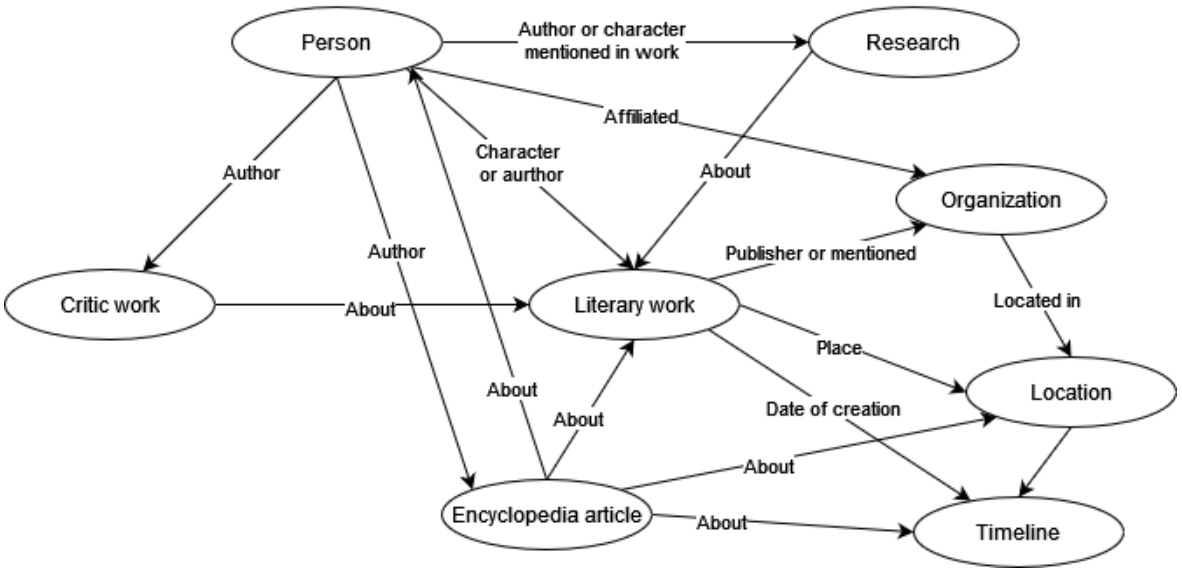


**Figure 1.** The proposed links between entities and texts.

Named Entity Recognition plays a crucial role in NLP and text analysis, where it is used to extract valuable information like people's names, locations, dates, organizations, and so on from the raw text [1]. The creation of the linking corpus for the writer's heritage could be based only on the extraction of common entities from the text and building relationships based on these entities, no matter which method is used. Its importance goes far beyond information extraction since NER is essential to many applications such as machine translation, text summarizing, information retrieval, and question-answering systems. Nowadays, with the huge amount of generated data, NER has become a crucial tool that will help academics, researchers, and organizations trying to analyze and extract the wealth of knowledge hidden in the textual corpus [2].

Research efforts in Named Entity Recognition focus mostly on the English language. Despite the growing interest in multi-language information extraction, few Russian-language studies report results. Although NER models for the English language have achieved impressive accuracy and usefulness levels, this is not necessarily the case for languages with less rich resources and morphological complexity like the Russian language. Building NER for the Russian language is considered to be more complex than with other languages because of the complexity and the unique rules the Russian language has [3], like its morphological complexity and the flexibility of the word order. In the Russian language, about all words like names, adjectives, and verbs change their forms and endings depending on the grammatical situation and context. Also, the flexibility of the word order allows us to write the same sentence with the same words in many different ways with only the word sequence varying. This makes understanding the context crucial to finding the semantic entities, not to mention that the Russian language and Russian literature have developed and changed significantly through the years, which makes building an

NER to determine and recognize the entities a challenging and complex task. As a result, there exists a growing demand for NER solutions tailored to Russian linguistic nuances and to their intended domains (like the custom NER we are trying to build for the A.S. Pushkin heritage).

To fill the gap between the available models and the desired solution, this research presents the creation of a unique NER system made especially for the A.S. Pushkin heritage. We use the Encyclopedia of A.S. Pushkin as the source of entities for training. This encyclopedia is an extensive resource devoted to Alexander Pushkin's life and contributions. It provides mentions and descriptions of all works by A.S. Pushkin, including letters and notes, persons related to him, places where he had been, and dates of the most important events. Therefore, it could be considered to be the most full source of knowledge about A.S. Pushkin's heritage. Furthermore, we evaluate the effectiveness and usefulness of this unique NER system in the field of Pushkin studies, shedding light on the usefulness of this system for academics and researchers to analyze Pushkin's literary and historical heritage.

The rest of the paper is structured as follows: Section 2 shows an overview of previous research related to our topic. Section 3 describes the dataset we used for training and evaluation. Section 4 talks about the steps we followed to prepare the dataset. Section 5 shows the training process and the results we achieved. Section 6 describes the database we established for storing and retrieving extracted entities. Section 7 concludes the paper and talks about future plans.

## 2. Related Work

In the field of natural language processing (NLP), the creation of customized Named Entity Recognition (NER) systems suited to particular linguistic and domain specifications has attracted significant interest. In this section, we examine previous research papers on the creation of NER systems for the Russian language, illuminating the difficulties and techniques that guide our research on building a unique NER system for the Encyclopedia of A.S. Pushkin.

There is an extension of the NER task that is aimed not only at the extraction of named entities but also at the creation of links from the entities in texts to the entities in some knowledge base. It is called the Named Entity Linking (NEL) problem. NEL could also be utilized for the creation of linked corpora. It can provide more accurate results since the goal of NEL is to find a direct link to an entity in the knowledge base from the text. In the case of the entity linking with the knowledge base, we don't need separate storage and entity verification. However, there is a very important limitation of the NEL problem. It fully relies on a knowledge base used to create links. If there is no appropriate entity in the knowledge base that could be linked with an entity in the text then the link will be lost, even if the entity is found in several texts and could be used to link them. It could be solved by the creation of a problem-specific knowledge base which contains all valuable entities from the problem domain, but it still needs to find all the entities with NER. Since we are working with a specific domain of A.S. Pushkin's literary heritage, our main goal is to find as many entities as possible to create links between the texts, and the NER is considered to be the better solution here.

The Russian language's complex grammatical structure, which includes inflection and a wealth of morphological information, creates special hurdles for NER. Previous publications have shown that researchers have used a variety of techniques to address these difficulties.

Many previous studies related to building Russian NER were introduced. A character-aware RNN model using LSTM units for Russian NER was able to identify the entities but struggled with distinguishing between person and organization tokens due to corpus size limitations [4]. The authors of [5] found that adding Conditional Random Fields (CRF) after the Bi-LSTM enhanced the performance of the NER when tested on the Gareev's, Person-1000, and FactRuEval 2016 datasets. Including CRF in a Bi-LSTM model enhances prediction quality when examining deep neural network models for Russian NER. The

effectiveness of multilingual NER models, which use information from other languages to improve NER in Russian, has been studied. In particular, those built on BERT seem to have the potential to improve NER in Russian. When comparing several pre-trained language models, the authors of [6] discover that Trankit-based models perform better on the NER challenge than others. Trails on BERT-based models trained on multilingual and Slovene-only data achieved high F-scores in Slovene and multilingual NER [7]. The authors of [8] presented a BERT model followed by a word-level CRF layer to address the problem of multilingual NER for the Slavic languages (including the Russian language). The presented model achieved the best results in the "BSNLP 2019 Shared Task" competition. The study [9] investigates the usage of a bidirectional BERT model that has been extensively trained to improve Russian Named Entity Recognition. The results are better than with a baseline BiLSTM network that includes additional layers, CNN, and FastText. Researchers explore how BERT models may benefit from cross-sentence information and show how adding context from several sentences can improve NER performance across a range of languages [10]. They show in their study [11] that performance in a variety of language-related activities is greatly enhanced when information from a multilingual model is transferred to a monolingual model.

One common strategy for obtaining good NER performance in the particular context of Russian text has been to modify pre-trained NER models to fit the unique characteristics of the Russian language. The authors of [12] presented a strong and effective fine-tuning approach that produced cutting-edge results on a range of NLP tasks. A few-shot fine-tuning framework for NER was presented by [13], demonstrating notable advancements over previous approaches. Their research [14] concentrated on low-resource NER challenges and introduced a pre-trained language model fine-tuning strategy that produced outstanding results on Hungarian and Uyghur datasets. Last but not least, the authors of [15] tackled the problem of zero-resource NER and suggested a target-oriented fine-tuning framework that produced cutting-edge outcomes on several benchmarks.

Annotated datasets are essential for Named Entity Recognition (NER) model training and evaluation, as demonstrated by a collection of previous studies. Annotation optimization techniques such as cautious annotations, consensus annotations, and multiple annotations are covered in [16]. The authors of [17] present a framework that combines knowledge-based techniques and neural models to provide high-quality multilingual silver data for NER. In their work [18], they investigate the usage of Wikipedia to automatically generate a huge corpus of annotated text called entities, which performs better in cross-corpus evaluation than the gold standard corpus. The authors of [19] focus on the cross-language transfer of high-quality annotations, using NER models with clinical texts written in low-resource languages through neural machine translation. These studies focus on how to improve the development of NER systems by creating a high-quality annotated corpus and effective annotation tools.

We acknowledge the significance of these discoveries in our study. We recognize that multilingual BERT models can improve our NER system for use with Pushkin's literary and historical heritage, and we take inspiration from their success. Our goal in fine-tuning such models is to create a customized NER system that balances the benefits of multilingual BERT with the particular requirements of the domain-specific material of Pushkin's literary and historical heritage. This method promises to provide an accurate and effective NER solution for the deep and varied world of Alexander Pushkin and his literary heritage. We are also looking to build a database that utilizes the NER system, allowing us to conduct a comprehensive and deep study of Pushkin's works. We will also utilize this database to perform an extended analysis of how the articles within the Encyclopedia of A.S. Pushkin are connected. Proceeding from that, we will build a knowledge tree of all the connected relationships between the extracted entities, like Pushkin's contemporary writers, their writings, dates of publication, etc.

## 3. Dataset Description

The dataset we used for training and evaluating our custom NER consists of four large volumes of Encyclopedia of A.S. Pushkin. Inside each volume, there are many text files holding information describing the previous works of the Russian writer Alexander Pushkin. This dataset contains very valuable and rich information devoted to the work of the writer and this encyclopedia was selected to include a wide range of information, such as in-depth narratives of Pushkin's literary works, biographies of writers who have written about these works, biographies of collaborators who worked with Pushkin, and additional details like dates, places, and organizations related to these literary projects.

The dataset is divided into four volumes as follows: Vol. 1 (228), Vol. 2 (277), Vol. 3 (310), Vol. 4 (174). The dataset contains 989 documents in total of Pushkin's works. Every text in these volumes captures a different aspect of the literary heritage of Alexander Pushkin. The wide range of content in these volumes—from contextual information to personal details and literary analyses—offers both gratifying and difficult ground for the creation of a unique NER system. Our goal is to improve the Encyclopedia of A.S. Pushkin's usability and accessibility by precisely extracting and classifying named entities through the careful annotation and training of an NER model on this dataset. This dataset is an invaluable tool for academics, researchers, and readers who want to understand the complex picture of Pushkin's life and literary accomplishments.

## 4. Dataset Annotation and Filtering

In this section, we talk about the process for preparing the dataset for training a custom NER. We describe in detail the steps we followed to filter, clean, and annotate the dataset. We shed light on the methods, models, and tools we used to achieve this purpose. We also deliver a comprehensive comparison of possible annotation tools and clarify our choices.

### *4.1. Annotating Tool*

In developing a custom Named Entity Recognition (NER) system for our dataset, choosing the best annotating tool is a crucial part and very important to facilitating the annotating process and increasing efficiency. So, we decided to conduct a comprehensive study of four annotation tools: Brat [20], NER Annotator [21], Tagtog [22], and Prodigy [23]. We will compare these tools, showing the advantages and disadvantages of each of them, focusing on choosing the best suitable tool for our task and explaining our choice. The following Table 1 describes the comprehensive comparison.

We chose to use the Brat annotation tool in our work since it has the ideal specifications for our work environment. Brat has a user-friendly interface which makes the annotating process easier for the annotating team and facilitates their work. Also, one of the biggest advantages of using Brat is its freeware and open-source nature. Brat can be deployed on our own web server, which has highly improved our workflow. This gave the annotating team a controlled environment and guaranteed data protection. The collaborative annotation capabilities of Brat were quite helpful for our research. Brat made it possible for several team members to work simultaneously on annotating the same dataset, which enhanced the annotation procedure and promoted productive collaboration. We were also able to customize the tool to meet the unique needs of the Encyclopedia of A.S. Pushkin dataset because of the flexibility and adaptability of the Brat tool, which ensured accurate and pertinent annotations.

**Table 1.** Annotation tools comparison: features and considerations.

| Annotating Tool | Advantages | Disadvantages |
| --- | --- | --- |
| **Brat** | **Open Source:** Brat is an open-source tool and free to use for annotation tasks.<br>**Web Server Deployment:** Bart can be deployed on a web server, which provides a safe and regulated setting for group annotation.<br>**Customization:** Gives the user freedom to define entity types and annotation policies. | **Lacks Machine Learning Integration:** Brat cannot integrate machine learning models or perform active learning.<br>**Initialization Difficulties:** Compared with some other tools, it is harder to set up the configuration files. |
| **NER annotator** | **User-Friendly:** An easy-to-use tool developed for non-technical users with a friendly User Interface design.<br>**Efficiency:** A simple and effective tool for NER annotation tasks. | **Limited Features:** For intricate or cooperative annotation tasks, NER Annotator might not provide all the functionality you need.<br>**Lacks Active Learning:** Neither machine learning model integration nor active learning is included in the NER Annotator tool. |
| **Tagtog** | **Collaboration Features:** Provides sophisticated tools for collaboration, enabling several annotators to collaborate on a single project.<br>**Data Preparation:** Reduces the process of preparing data and can aid in the training of machine learning models. | **Paid Service:** A paid membership may be needed for some services, which makes it less affordable for large-scale projects with tight budgets.<br>**Complexity:** Advanced features need technical experience to utilize. |
| **Prodigy** | **Active Learning:** Utilizes active learning strategies that enable the creation of highly accurate annotation models quickly and with less effort.<br>**Scalability:** It is suitable for small-scale and large-scale annotation applications. | **Cost:** Prodigy is a paid tool, which makes it not suitable for some applications.<br>**Complexity:** For users who are unfamiliar with machine learning integration, utilizing this tool may be difficult. |

### *4.2. Annotation Process*

The annotating process for building custom NER for the Encyclopedia of A.S. Pushkin is a multi-stage process containing crucial steps for clearing, filtering, and preparing the dataset. In this subsection, we will delve into each step we followed to prepare the dataset and shed light on the methods and techniques we used. Figure 2 shows the pipeline for the process of data preparation:

1. **Proofreading and Document Structuring**: The first step in preparing the dataset is having domain experts examine the raw dataset. Every single text within the dataset undergoes a thorough proofreading and review process to improve its quality and readable content. Text files are also separated into organized paragraphs to make processing steps easier.
2. **Data Cleaning and Formatting**: To make sure that the textual data can be analyzed by NER models, a data cleaning and filtering phase was carried out. This involves removing unnecessary punctuation, white spaces, and misspelled characters. The target of this stage is to deliver clear, clean, and organized texts that will enhance performance when training the NER system.
3. **Annotation Using Pre-trained NER Model**: Regarding the NER annotation procedure, an advanced Russian pre-trained NER model, such as DeepPavlov [24], is utilized. Named entities in the text can be recognized and annotated using this model. In particular, the annotated entity types are as follows:
    - **PER (Person)**: Recognizing people who are referenced in the book, such as historical figures, writers, critics, and commentators.
    - **DATE (Date)**: Identifying all the entities that give the text a chronological framework, such as periods and dates.
    - **LOC (Location)**: Identifying the regions and locales that are mentioned in the text, such as cities and countries.

- **ORG (Organization)**: Identifying associations and communities mentioned in the text, such as companies and institutions.
- **WORK-OF-ART (Work of Art)**: Recognizing literary works and manuscripts within the text, such as books' titles and poems' titles.

4. **Entity Correction and Alignment**: After using DeepPavlov to annotate the text, reviewing and editing the annotated entities is a crucial step that should be performed. To make sure they are precisely aligned with the matching text spans, we closely inspected the annotated entities. The entity boundaries are checked for accuracy and contextual relevance, and adjustments are made as necessary.
5. **Regular Expression Annotator for "WORK-OF-ART" Entities**: To compensate for the potential poor performance of the DeepPavlov pre-trained NER model, a unique regular expression annotator is constructed for the "WORK-OF-ART" entity. This guarantees the proper recognition and annotation of Pushkin's literary works inside the text, which might vary greatly and rely on context.
6. **Verification by Domain Specialists**: Domain experts carry out a thorough verification procedure on the final annotated dataset. These specialists examine the annotated entities concerning the documents' context to make sure that the annotations appropriately capture the semantic meaning of the entities and are consistent with the underlying material. This step involves identifying and fixing any inconsistencies or errors.

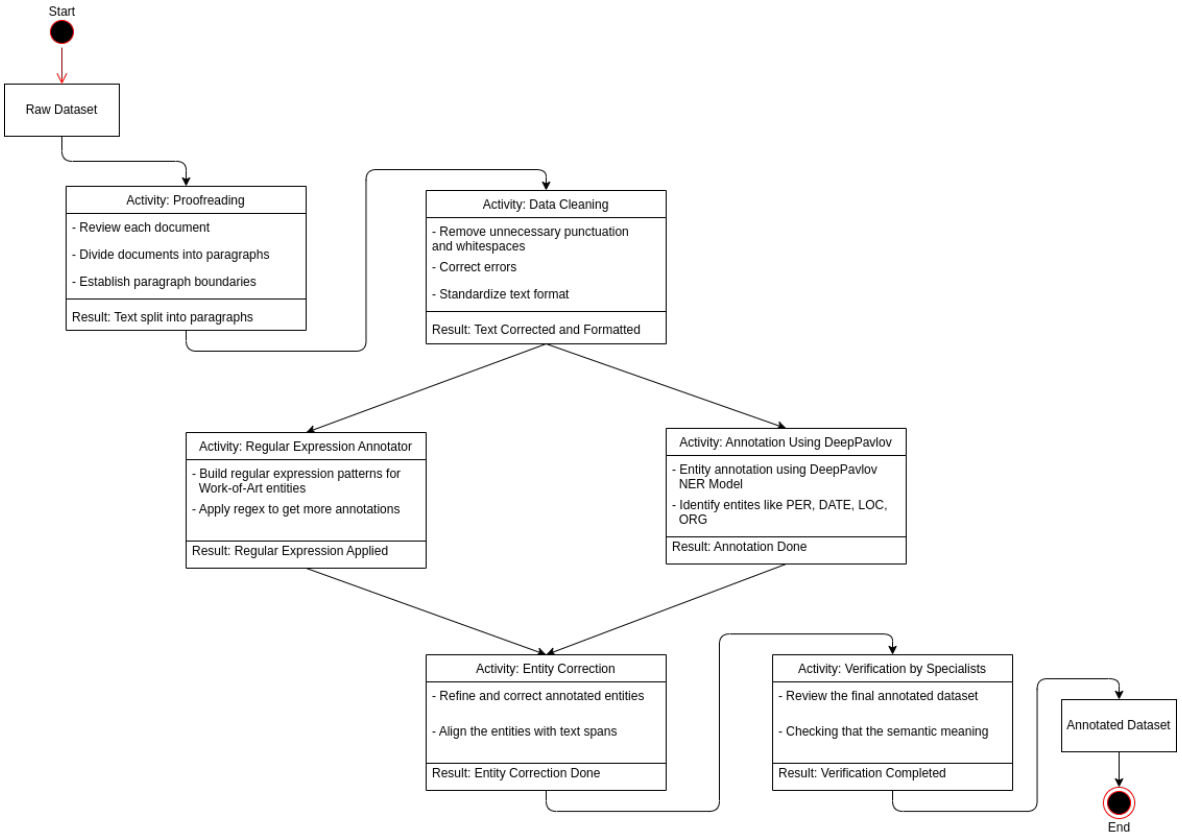


**Figure 2.** The pipeline of annotating the dataset.

Following the previous steps we explained in detail, we are focusing on building a high-quality dataset that can be used for the purpose of training and deploying a custom NER system for the Encyclopedia of A.S. Pushkin. Along with improving entity recognition accuracy, this dataset guarantees that the NER system can handle the complex and domain-specific content found in the Encyclopedia of A.S. Pushkin, which will

greatly assist academics, researchers, and readers alike as they delve deeper into Pushkin's literary heritage.

## 5. Model Training

In this section, we dive into the process of training our custom NER models for the Encyclopedia of A.S. Pushkin. The SpaCy [25] framework was selected to train our models because of its effectiveness and strong NER capabilities. The training process can be described as follows:

- **Data Preparation**: One of the most important things we had to do during our project was to convert our annotations into different formats to make them compatible with the models and tools that we were using. Initially, we used DeepPavlov to annotate our dataset. Processing from that, we had to convert the annotations to ".ann" format so they could be used in the Brat annotation tools, which allowed the annotation team to collaborate on fixing and adjusting the annotated entities. After that, the annotations were then converted from ".ann" to ".json" format. This transformation was needed to make sure the annotated dataset was perfectly compatible with the SpaCy models for fine-tuning. This procedure made sure that our annotated dataset was correct and complete, as well as prepared for the subsequent stages of model building and training.
- **Fine-tuning with Pre-trained Transformer Model**: We utilized the powerful structure of SpaCy to train our custom NER model. SpaCy is a great option for our goal as it offers an extensive collection of tools for training NER models. We optimized a BERT-based multilingual transformer model that had already been trained using pre-learned contextual embeddings inside the SpaCy framework. This improved the model's performance when applied to text in Russian. To handle the linguistic difficulties of the Russian language, it is useful to use a BERT base transformer model that has already been trained. The pre-trained model serves as an excellent starting point, capturing a wide range of linguistic patterns and context, which can be fine-tuned to our specific dataset.

SpaCy is the best option for our purpose due to its NER capabilities, speed, and efficiency. It guarantees that our unique NER model is correct and suitable for production. It also offers annotated formats that are suitable for our final production, like ".xml" and ".html" formats.

We chose multilingual BERT transform models because of their higher performance in NLP tasks, especially in NER tasks, as we highlighted in the Related Work section. These models accurately represent the contextual subtleties of the target language. By taking into account the surrounding words and phrases, they are excellent at identifying named entities.

Using a BERT-based multilingual transformer model, we have been able to fine-tune the model to our task. Since these models are multilingual, they can be adjusted to fit the many linguistic contexts found in Pushkin's writings. They can identify words and phrases from Russian and other languages that could be used in Pushkin's writings. These models can be fine-tuned to a specific dataset, which is a crucial factor to maximize the performance of our NER. We also enhanced the model's performance by providing high-quality annotated entities achieving that through special techniques such as the regular expression annotator for "WORK-OF-ART". By following these steps, we were able to train a custom NER that can identify the entities in the Russian heritage of Alexander Pushkin. The training process explained in this study can be generalized to build a similar NER for other literary works while taking into account the uniqueness of each language's literature.

We made multiple training experiments and the best results we achieved using each architecture are shown in the following tables. Table 2 describes the model we used for training and also shows the situation of the dataset we used. Table 3 shows the training results we achieved on the test set.

**Table 2.** Description of the NER models for processing A.S. Pushkin's heritage.

| Model | Description |
|---|---|
| Model #1 | Tok2vec model to train on auto-annotated dataset before adjusting the work-of-art entities. |
| Model #2 | Transformer-based model to train on auto-annotated dataset after adjusting the work-of-art entities. |
| Model #3 | Tok2vec model to train on the dataset version checked by specialists. |
| Model #4 | Transformer-based model to train on the dataset version checked by specialists. |

**Table 3.** Results of training multiple NER models with scores corresponding to each entity type.

| Model | Comments | Entity | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Model #1 | Tok2vec model before adjusting the work-of-art entities | Ents | 0.6202 | 0.6466 | 0.6331 |
| | | PER | 0.7804 | 0.8276 | 0.8033 |
| | | LOC | 0.5337 | 0.5471 | 0.5403 |
| | | WOA | 0.1103 | 0.0947 | 0.1019 |
| | | DATE | 0.7280 | 0.9032 | 0.8062 |
| | | ORG | 0.3272 | 0.3333 | 0.3302 |
| Model #2 | Transformer-based model after adjusting the work-of-art entities | Ents | 0.7718 | 0.7806 | 0.7761 |
| | | PER | 0.7994 | 0.8214 | 0.8102 |
| | | LOC | 0.5361 | 0.5597 | 0.5476 |
| | | WOA | 0.8637 | 0.6987 | 0.7725 |
| | | DATE | 0.7306 | 0.8930 | 0.8037 |
| | | ORG | 0.3142 | 0.2868 | 0.2962 |
| Model #3 | Tok2vec model trained on the checked dataset | Ents | 0.8742 | 0.8367 | 0.8550 |
| | | PER | 0.8850 | 0.8674 | 0.8761 |
| | | LOC | 0.6412 | 0.5283 | 0.5793 |
| | | WOA | 0.9030 | 0.9114 | 0.9114 |
| | | DATE | 0.8764 | 0.7880 | 0.8299 |
| | | ORG | 0.7352 | 0.4629 | 0.5681 |
| Model #4 | Transformer-based model trained on the checked dataset | Ents | 0.9020 | 0.8885 | 0.8952 |
| | | PER | 0.9357 | 0.9004 | 0.9177 |
| | | LOC | 0.8478 | 0.7358 | 0.7878 |
| | | WOA | 0.8577 | 0.9083 | 0.8823 |
| | | DATE | 0.9077 | 0.8909 | 0.8992 |
| | | ORG | 0.6949 | 0.7592 | 0.7256 |

As we can see from the previous table, our custom NER model achieved very good results on the test set. It archives an average F1-score of 0.8952 across the five entity types (PER, LOC, ORG, DATE, WOA). These results are considered to be very good when taking into account the fact that we only used the first volume of the encyclopedia for training (80% of the text files) and testing (20%).

## 6. Database Creation and Structure

Creating an organized and effective database to handle the extracted entities was a crucial part of the development of an approach to linked corpus creation over the literary heritage of the A.S. Pushkin. In this section, we talk about the database we created that utilized the NER model we trained to store the extracted entities from the textual data. The database we built contains the following tables, where Figure 3 shows a scheme for the database:

- **Main Table**: This table contains all the entities extracted from the dataset by utilizing the NER model we trained. Each entity within the table has a unique ID as well as information that describes the entity type, entity span text, and text from which it was extracted.
- **Separate Tables for Entity Types**: To optimize efficiency and the structure, we created distinct tables for every kind of entity: Persons (PER), Organizations (ORG), Locations

(LOC), and Works of Art (WOA). These separate tables ensure an organized and efficient storage solution. Inside each entity type table, we store the unique entities we found in the text under this entity type.

- **Link Tables**: To achieve connections between the main tables that contain all the extracted entities and the separate tables for each entity type, we built these link tables. These tables act as connectors between entities and their corresponding entity type tables. These link tables help to create a complete and integrated dataset.

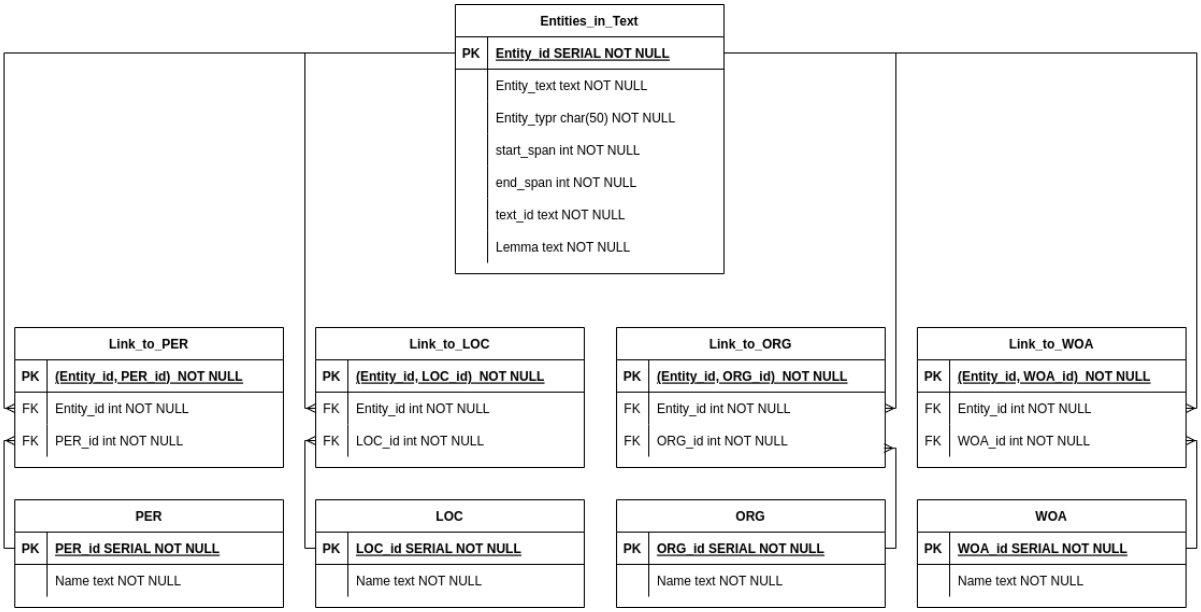


**Figure 3.** The database scheme.

The importance of building this database can be summarized as follows:

- **Efficient Storage and Retrieval**: This database will allow the user to easily and effectively store the extracted entities while maintaining a well-organized form. The structured database will allow for rapid access for entities and the rapid extraction of information.
- **Facilitating Future Enhancements**: The database will act as a fundamental structure when we add more data to the dataset—like external URLs—over time. This gives users the freedom to easily add more data and expand the database. It will make the expansion of the stored information easier for future adjustments.
- **Enhancing User Experience**: Users can do thorough searches in this database to find relevant data linked to a certain entity. Users may quickly access an enormous amount of knowledge on people, places, businesses, and artistic creations listed in the Encyclopedia of A.S. Pushkin by doing database queries. The database facilitates in-depth investigation and study while greatly enhancing the user experience.

In a nutshell, the well-organized database we built plays a crucial part in the approach to creating a linked corpus, facilitating and enhancing the process of storing, retrieving, and searching for entities. Also, this database is considered to be a base for building more advanced applications, such as a web service that utilizes this database to enhance the user experience.

## 7. Conclusions

In this research, an approach to building a linked corpus over the literary heritage of the great Russian poet A.S. Pushkin is proposed. The approach is based on the Named Entity Recognition method with an analysis of the unique entities in different texts. The multilingual BERT model is optimized to build a custom NER for Russian texts and is used to extract various semantic entities within literary works. The process of filtering and preparing the text dataset for training and evaluating the NER model is explained. We

have highlighted the important methods and techniques we used to improve the quality of the dataset we used. The results achieved by our trained NER models are described, which showed very good results in extracting and identifying named entities in the test set. The database we created to store entities extracted from texts is discussed, which deeply demonstrates the importance of this database in advancing our research. The NER system and database play a crucial role in building a linked corpus through various works within Pushkin's literary legacy.

The NER system and the database we built provide significant support for researchers and academics working in the digital humanities and on various literary works. It helps them to analyze the literature more efficiently by extracting valuable information from these works, not to mention how it reduces the time needed to examine and study these works. Also, the database with the linked entities provides a deeper and more comprehensive view of how the multiple literary works are connected through their common entities.

Future work will focus on enhancing the NER results by increasing the size of the dataset and exploring other techniques that might maximize the performance of the NER on the A.S. Pushkin heritage. Next, we are going to provide an additional estimation of the created dataset and model trained with it as well as a comparison of the model with existing models. The additional direction of our future work is a developing approach to providing links from the found entities to a Wikidata knowledge base as a part of NEL problem solving. We are also planning to build a web service that utilizes our trained NER models and the database we built to make a more efficient and easier way for the user to access the rich information within Pushkin's works.

**Author Contributions:** K.K. was responsible for building and training the NER system, creating the database that utilizes the NER system, and writing the paper. N.T. was responsible for deploying the annotating tool on the server, mentoring the annotation process, paper editing, and peer review. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the State Research FFZF-2023-0001.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable

**Data Availability Statement:** The data are not publicly available now due to the privacy usage by the project's funders. The data might be available in the future.

**Acknowledgments:** The authors would like to thank colleagues from the Institute of Russian Literature of the Russian Academy of Sciences for proofreading texts from the Encyclopedia of A.S. Pushkin and for their invaluable contribution in the verification of NER results.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Mansouri, A.; Affendey, L.; Mamat, A. Named Entity Recognition Approaches. *Int. J. Comp. Sci. Netw. Sec.* **2008**, *8, 339–344.*
2. Li, J.; Sun, A.; Han, J.; Li, C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 50–70. [CrossRef]
3. Gareev, R.; Tkatchenko, M.; Solovyev, V.D.; Simanovsky, A.; Ivanov, V. Introducing Baselines for Russian Named Entity Recognition. In Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics, Samos, Greece, 24–30 March 2013 .
4. Malykh, V.; Ozerin, A. Reproducing Russian NER Baseline Quality without Additional Data. In Proceedings of the CDUD@CLA, Moscow, Russia, 18–22 July 2016.
5. Lê, T.A.; Arkhipov, M.; Burtsev, M. Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition. In *Artificial Intelligence and Natural Language: 6th Conference, AINL 2017, St. Petersburg, Russia, September 20–23, 2017, Revised Selected Papers 6*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 91–103.
6. Suppa, M.; Jariabka, O. Benchmarking Pre-trained Language Models for Multilingual NER: TraSpaS at the BSNLP2021 Shared Task. In Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing—Association for Computational Linguistics, Kiyv, Ukraine, April 2021; pp. 105–114.
7. Prelevikj, M.; Žitnik, S. Multilingual Named Entity Recognition and Matching Using BERT and Dedupe for Slavic Languages. In Proceedings of the Workshop on Balto-Slavic Natural Language Processing, Kiyv, Ukraine, April 2021; pp. 80-–85.

8. Arkhipov, M.; Trofimova, M.; Kuratov, Y.; Sorokin, A. *Tuning Multilingual Transformers for Language-Specific Named Entity Recognition*; Association for Computational Linguistics: Kerrville, TX, USA, 2019; pp. 89–93. [CrossRef]
9. Mukhin, E. *Using Pre-Trained Deeply Contextual Model BERT for Russian Named Entity Recognition*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 167–173. [CrossRef]
10. Luoma, J.; Pyysalo, S. Exploring Cross-sentence Contexts for Named Entity Recognition with BERT. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 1 December 2020; pp. 904–914. [CrossRef]
11. Kuratov, Y.; Arkhipov, M. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. *arXiv* **2019**, arXiv:1905.07213.
12. Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; Zhao, T. SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 2177–2190. [CrossRef]
13. Wang, Z.; Zhao, K.; Wang, Z.; Shang, J. Formulating Few-shot Fine-tuning Towards Language Model Pre-training: A Pilot Study on Named Entity Recognition. *arXiv* **2022**, arXiv:2205.11799. https://doi.org/10.48550/arXiv.2205.11799.
14. Chen, S.; Pei, Y.; Ke, Z.; Silamu, W. Low-Resource Named Entity Recognition via the Pre-Training Model. *Symmetry* **2021**, *13*, 786. [CrossRef]
15. Zhang, Y.; Meng, F.; Chen, Y.; Xu, J.; Zhou, J. Target-oriented Fine-tuning for Zero-Resource Named Entity Recognition. *arXiv* **2021**, arXiv:2107.10523.
16. Grouin, C.; Lavergne, T.; Névéol, A. Optimizing annotation efforts to build reliable annotated corpora for training statistical models. In Proceedings of the LAW VIII—The 8th Linguistic Annotation Workshop, Dublin, Ireland, August 2014; pp. 54–58. [CrossRef]
17. Tedeschi, S.; Maiorca, V.; Campolungo, N.; Cecconi, F.; Navigli, R. WikiNEuRal: Combined Neural and Knowledge-based Silver Data Creation for Multilingual NER. In Findings of the Association for Computational Linguistics: EMNLP, Punta Cana, Dominican Republic, 11 November 2021. [CrossRef]
18. Nothman, J.; Murphy, T.; Curran, J.R. Analysing Wikipedia and Gold-Standard Corpora for NER Training. In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Athens, Greece, March 2009; pp. 612–620.
19. Schäfer, H.; Idrissi-Yaghir, A.; Horn, P.; Friedrich, C. Cross-Language Transfer of High-Quality Annotations: Combining Neural Machine Translation with Cross-Linguistic Span Alignment to Apply NER to Clinical Texts in a Low-Resource Language. In Proceedings of the 4th Clinical Natural Language Processing Workshop, Online, 14 July 2022. [CrossRef]
20. Stenetorp, P.; Pyysalo, S.; Topic, G.; Ohta, T.; Ananiadou, S.; Tsujii, J. brat: A Web-based Tool for NLP-Assisted Text Annotation. In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 2012.
21. Arunmozhi.; Khan, A.; Kunert, L. Ner-Annotator. Available online: https://github.com/tecoholic/ner-annotator (accessed on 12 May 2023).
22. Pierto, C.; Cejuela, J. M. TagTog: The Text Annotation Tool to Train AI. Available online: https://docs.tagtog.com/ (accessed on 20 October 2023).
23. Montani, I.; Honnibal, M. Prodigy: An annotation tool for AI, Machine Learning. Available online: https://prodi.gy/ (accessed on 22 October 2023).
24. Burtsev, M.; Seliverstov, A.; Airapetyan, R.; Arkhipov, M.; Baymurzina, D.; Bushkov, N.; Gureenkova, O.; Khakhulin, T.; Kuratov, Y.; Kuznetsov, D.; et al. DeepPavlov: Open-Source Library for Dialogue Systems. In Proceedings of the ACL 2018, System Demonstrations, Melbourne, Australia, 15–20 July 2018. [CrossRef]
25. Honnibal, M.; Montani, I.; Van Landeghem, S.; Boyd, A. spaCy: Industrial-Strength Natural Language Processing in Python. 2020. Available online: https://zenodo.org/records/10009823 (accessed on 22 October 2023).