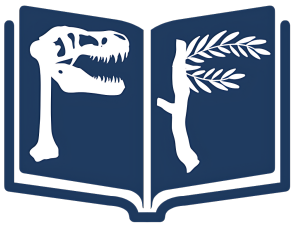
Final Report:
Finding Fossils in the Literature



In partnership with the Neotoma Paleoecology Database.

Ty Andrews Jenit Jain Kelly Wu Shaun Hutchinson

2023-06-26

Executive Summary

The project Finding Fossils in Literature is sponsored by the Neotoma database (Neotoma) which houses a paleoecology database. The challenges Neotoma faces are 1) researchers have to manually enter sample data into Neotoma, 2) researchers are not aware of Neotoma or that their research fits into it, and 3) there are too many articles published for the Neotoma team to monitor new research. This project has 3 primary deliverables to solve the challenges, first is an article relevance prediction model which predicts whether newly published articles are relevant to Neotoma. Second, is an article data extraction pipeline which identifies key entities such as taxa or geographic location. Last is a data review tool for Neotoma data stewards to review the extracted data before it is submitted to Neotoma. Once the extracted data has been reviewed, the corrections will be used to retrain the first two models of the pipeline.

Table of contents

1 Introduction

The Neotoma database (Neotoma) (Williams et al. 2018) serves as a valuable resource for researchers investigating ecological transformations spanning the last 5 million years. Nevertheless, the collection of data heavily relies on manual submissions from researchers, presenting challenges in data entry and impeding collaborative endeavors aimed at achieving a comprehensive understanding of ecological changes.This project aims to automate the extraction of data from relevant journal articles which can be added to Neotoma. This completed in three parts. First, article relevancy to Neotoma is predicted. Relevant articles are then parsed using natural language processing (NLP) techniques. Finally, an interactive data review tool was built to review and correct the extracted data before it is submitted to Neotoma. The outputs of this data review tool are then used to retrain the two data models in the future.

2 Data Science Methods

2.1 Article Relevance Prediction

The article relevance prediction is posed as a binary classification problem. Based on the available article metadata, the model is able to predict how likely the article is relevant to the Neotoma paleoecology/paleoenvironment database. If the article is deemed likely relevant, the article will proceed to the next stage for article data extraction.

2.1.1 Approach

2.1.1.1 Building the Training Data

To train the supervised classification model, a sample of labelled articles has been compiled as shown in Table 1.

Table 1: Labelled Sampled Article

Label	Sample Size	Description
Positive	911	Through the assistance of Neotoma, a randomly sampled list of articles that currently contribute to Neotoma was provided as the positive cases. These articles cover various types of fossil data in the Neotoma database.
Negative	3523	Articles from non-paleoecology-related subjects were queried to form a representative sample of non-relevant articles. Among these articles, there are articles that are closely related to paleoecology but do not contain fossil collection data.

The labelled articles were split into train/validation/test sets with splits of 0.7/0.15/0.15, which correspond to 3103/665/666 articles. Due to the relatively small total sample size, 70% of the articles were included in the training set so that there are more examples provided during training.

2.1.1.2 Preprocessing & Feature Selection

Article's metadata were retrieved from the CrossRef API. There were over 50 types of metadata provided by the CrossRef API (Crossref 2023). Since many of them are related to the publication aspect of the article, not the contextual aspect, feature selection was performed

to reduce dimensionality. Table 2 explains the key decisions made during the feature selection step.

Table 2: Feature Selection Decisions

Feature	**Decision**
Title & subtitle	Title and subtitle (if any) are concatenated as a descriptive text feature. Text representation techniques were then applied to this feature.
Abstract	Due to copyright issues, less than half of the articles have an abstract available from the CrossRef API. However, the abstract contains valuable information about the article's content. Thus, to solve the missing data problem when the abstract is available, it is concatenated with the article title and subtitle. Altogether they formed a descriptive text feature. Considering that the availability of an abstract could affect model prediction, a binary feature was derived to indicate if the article's text feature has an abstract or not.
Author & Journal	In order to mitigate potential bias towards well-established authors and journals, the feature list was modified to exclude time-sensitive indicators so that it could account for new Authors and Journals published in the future.
Subject	Instead of using the name of the journal, the subject of the journal was used as a proxy of the journal.
Number of citation	Number of citations could potentially indicate if the article contains important data and is referred to in other articles.
Feature related to the publication process	These features do not provide any contextual information about the article, thus they were removed from the feature list.

The features selected include a descriptive text feature (concatenation of title, subtitle and abstract), subject of the journal, number of citations and a binary indicator for having an abstract available from CrossRef or not.

2.1.1.3 Feature Engineering: Text representation

The descriptive text feature provides crucial contextual information for the model to predict if the topic is related to paleoecology or not. Feature engineering was conducted to represent this descriptive text feature in numeric forms. The following text representations were explored and tested to identify the most effective approach.

Bag of words representation (Baseline): word-frequency based method. While the model with bag of words feature achieved a decent performance, the limitation is that the semantic aspect of the text could not be well represented. This method was chosen to be used in the baseline model.

Adding term-association probability with zero-shot classification (Not Used): providing a candidate label, a transformer based model could predict the probability that the text is related to the candidate label. The idea was to let the pre-trained model add the text association probability for several paleoecology related terms. This method was ultimatly not used because not only the handcrafted list of terms could introduce subjective bias in the model, but also the added term-association feature did not show high feature importance in the trained models.

Sentence embedding (Final): Sentence embedding is the state-of-the-art approach for text representation. Three sentence embedding models were experimented and allenai/specter2 model resulted in the best overall performance as documented in Table 3.

Table 3: Sentence Embedding Models

Model	Result
bert-tiny-finetuned-squadv2	This model is of a small scale so that the embedding process will be quick.
Biobert	This model is pre-trained based on large-scale biomedical corpora, thus it could be better at understanding biology-related jargons that frequently appear in the field of paleoecology.
specter2	This model is pre-trained using over 6 million scientific articles, thus it could be better at understanding the language style and pattern in scientific articles.

2.1.1.4 Model Selection

The following selection of Probability Classifiers, Analogy-based and Tree-based & Gradient-boosted models were experimented with.

Probability classifiers: Traditional probability classifiers provide fast training and prediction time, which is a plus when processing and retraining hundreds and thousands of new

articles. Its simplicity also provide good model transparency and interpretability for model evaluation. Logistic regression and naive Bayes were experimented during model selection.

Analogy-based models: The nature of the article relevance prediction problem could be framed as looking for articles that are similar to the positive examples. Analogy-based models can be suitable for this problem. K-nearest neighbour and Support Vector Machine with RBF kernel were experimented.

Tree-based & Gradient-boosted models: Tree-based classifiers could use different feature subsets at various steps during the prediction, which can be suitable for our relatively high dimensional data problem. Decision Tree, Random Forest Model, LightGBM, XGBoost and CatBoost were experimented.

The choice of the final model was primarily based on model performance on recall score. The chosen model went through further hyperparameter tuning to optimise the performance.

2.1.2 Model Evaluation

Recall score is the main metric to optimise. The main goal of article relevance prediction is to identify potentially as many useful articles as possible from the giant article repository so that valuable research data can be discovered. The cost of false negatives is high because if the model did not catch a relevant article, this article will unlikely to be discovered by the Neotoma community and its data will be missed. On the other hand, the cost of false positives is more manageable. During data validation, Data Stewards can easily mark an article as not relevant using our data review tool.

Another key consideration is model interpretation. The positive examples were gathered using articles that currently exist in the Neotoma database, and it is difficult to tell if there are existing biases in the training data. A more interpretable model would help identify model biases and make corrections in the long run.

2.2 Article Data Extraction

The article data extraction is posed as a Named Entity Recognition (NER) problem. The NER tasks teaches the model to predict the correct label for each word where each word is assigned a label.

2.2.1 Data Description

An original sample of 300+ full text articles related to Neotoma was provided. Some articles were non-english and were removed from the dataset. The data of interest to be extracted and thus labelled in the articles using NER are as follows:

- **SITE**: the specific site name given to the items studied
- **REGION**: general geographic regions around the site to give general context to locations
- **GEOG**: geographic coordinates of sites or samples
- **TAXA**: the taxa uncovered and researched in the article
- **AGE**: any ages relevant to the samples
- **ALTI**: the altitude at which samples were studied
- **EMAIL**: any researchers emails for further follow up

2.2.2 Data Preprocessing and Labelling

For the development of the NER models a sample of 39 articles relevant to Neotoma were preprocessed and labelled by the research team using LabelStudio. Preprocessing steps involved chunking the articles, varying in length drastically, into bite-sized pieces and replacing special tokens with their English equivalent symbol. Other text preprocessing techniques like stemming, lemmatization and converting text to lower case were not used for this project as the aim was to preserve the original text and ensure that the model understands contextual information. The Label Studio interface hosted on HuggingFace was used for labeling purposes as it provides an out-of-the-box collaborative environment for several ML tagging tasks, including named entity recognition. Input and output files were efficiently synced with buckets on the Azure cloud, with scope of onboarding a larger team in the future to carry on model development. The SITE entity proved difficult to label as differentiating between obscure location names or an acronym of a core sample was difficult for the team. Similarly, correctly identifying TAXA and where on TAXA label began and another ends proved challenging. A significant improvement in results is expected by improving the quality of labeling with the help of data stewards with subject matter expertise.

The labelled articles were split into train/validation/test sets by full article with splits of 0.7/0.15/0.15 respectively. The data splitting was done based upon whole articles to prevent data leakage into the model learning certain articles patterns before evaluation. This resulted in the final dataset with the statistics as shown in Figure 1.

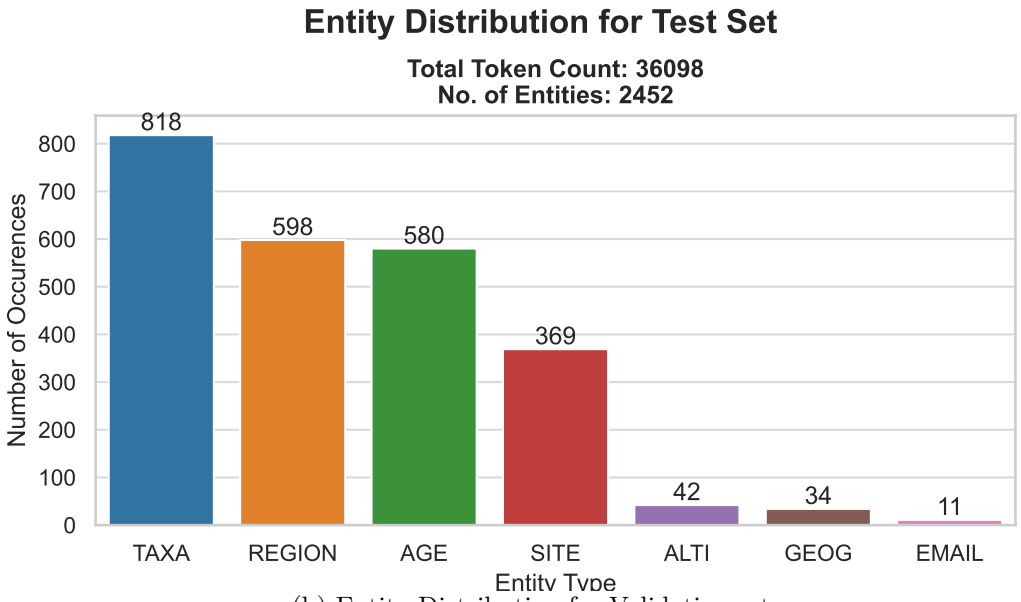
2.2.3 Approach

The state of the art for NER tasks is currently dominated by transformer based models. The best performing non-transformer based models are a transition-based stack long-short term memory (S-LSTM) model used in the spaCy package.

The following approaches were considered along with the rationale for their inclusion/rejection from development.

Figure 1: Entity Distribution

(a) Entity Distribution for Test set



(b) Entity Distribution for Validation set

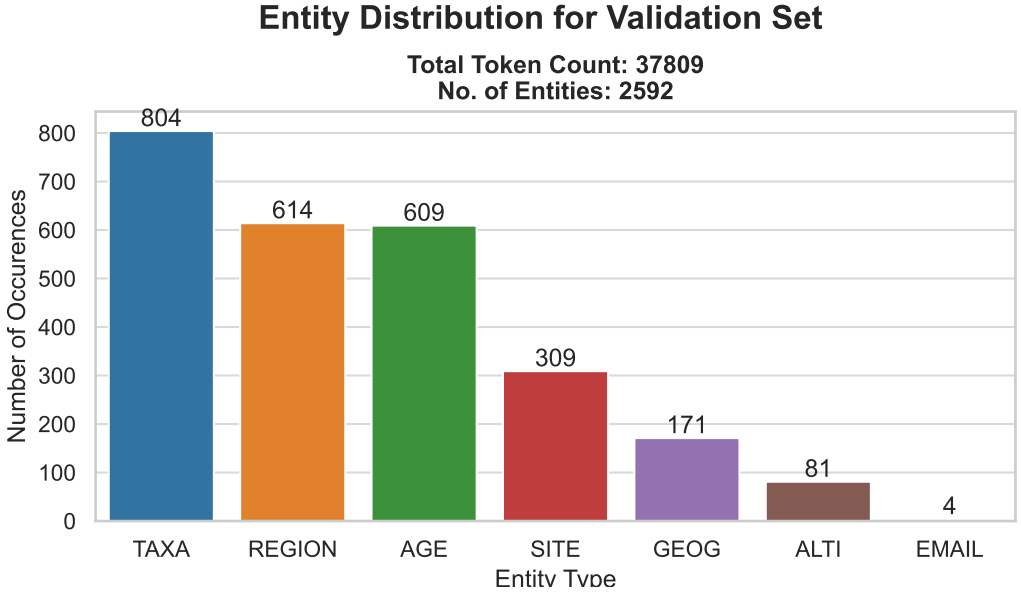


Table 4: NER Approaches

Rule Based Models	This served as the baseline using regex to extract known entities but was not developed further due to the known issues with text quality due to OCR issues and infeasibility for entities like SITE.
Transition Based Model	Developed as a lower computational complexity solution with potential for low-compute resource deployment applications
BERT Based Models	These are achieving state of the art results and have multiple base models pre-trained on different domains making them an attractive option.
Text to Text Transfer Transformer(T5)	T5 models turn every problem into a text to text problem which for NER requires significant pre/post processing to work and was excluded for this reason.

For the transformer based models two approaches were used for training, spaCy CLI and Hugging Face's Training API. Both come with the following advantages and disadvantages:

Table 5: spaCy CLI andHugging Face's Training API Advantages and Disadvantages

	Pro	Con
spaCy Config Training	• can integrate with any transformer hosted on HuggingFace • Prebuilt config scripts that require minimal changes • Access to spaCy's unique transition-based NER model • Ability to integrate multiple ML models in an ensemble -Integrates with visualization libraries	• Knowledge of bash scripting required • Limited configuration options
Hugging Face Trainer API	• able to use non-NER models and add classification head easily • able to easily integrate custom tracking of metrics and external logging to MLflow	• more code required

Using the Hugging Face training API the following models were trained and evaluated along with the hypothesis behind their selection.

Table 6: Hugging Face Model Hypotheses

Model	Hypothesis
RoBERTa-base	Cited as one of the commonly best performing models for NER [SOURCE]
RoBERTa-large	A larger model than the base version with potential to learn more complex relationships with the downside of larger compute times.
BERT-multilanguage	The known OCR issues and scientific nature of the text may mean the larger vocabulary of this multi-language model may deal with issues better.
XLM-RoBERTa-base	Another cross language model (XLM) but using the RoBERTa base architecture and pre-training.
Specter2	This model is BERT based and finetuned on 6M+ scientific articles with it's own scientific vocabulary making it well suited to analyzing research articles.

Final hyper parameters used to train the models were as follows:

Table 7: Hugging Face Hyperparameters

Parameters	Notes
Batch size	• Maximized to utilize all available GPU memory, 8 for RoBERTa based models, 4 for large models
Gradient Accumulation	• Used to mimic larger batch sizes, this value was chosen to achieve batch sizes of ~12k tokens based on existing research [SOURCE]
Epochs	• Initial runs with 10-20 epochs, observed evaluation loss minima occurring in first 2-8 depending on learning rate below
Learning Rate	• Initially 5e-5 was used and observed rapid over fitting with eval loss reaching a minimum around 2-4 epochs then increasing for the next 5-10 • Moved to 2e-5 as well as introducing gradient accumulation of 3 epochs to increase effective batch size, the eval loss didn't reach a minimum for a bit longer while recall continued to improve
Learning Rate Scheduler	• All initial training has been done with a linear learning rate scheduler which linearly decreases learning rate across epochs
Warmup Ratio	• How many steps of training to increase LR from 0 to LR, shown to improve with Adam optimizer - (AI 2023) Set to 10% initially |

Using the spaCy CLI, the following models were trained and evaluated along with their pros and cons highlighted.

Table 8: spaCy CLI Model Advantages and Disadvantages

Model	Advantages	Disadvantages
RoBERTa-base	• State-of-the-art pretrained transformer for NLP tasks in English • Context rich embeddings	• Computationally expensive to train and inference • Cannot fine-tune • Slow processing
en_core_web_md	• A smaller word vector model with static embeddings for words • Memory and speed efficient	• Static embeddings without context • Cannot handle OOV words well • Cannot fine-tune

Final hyper parameters used to train the models were as follows:

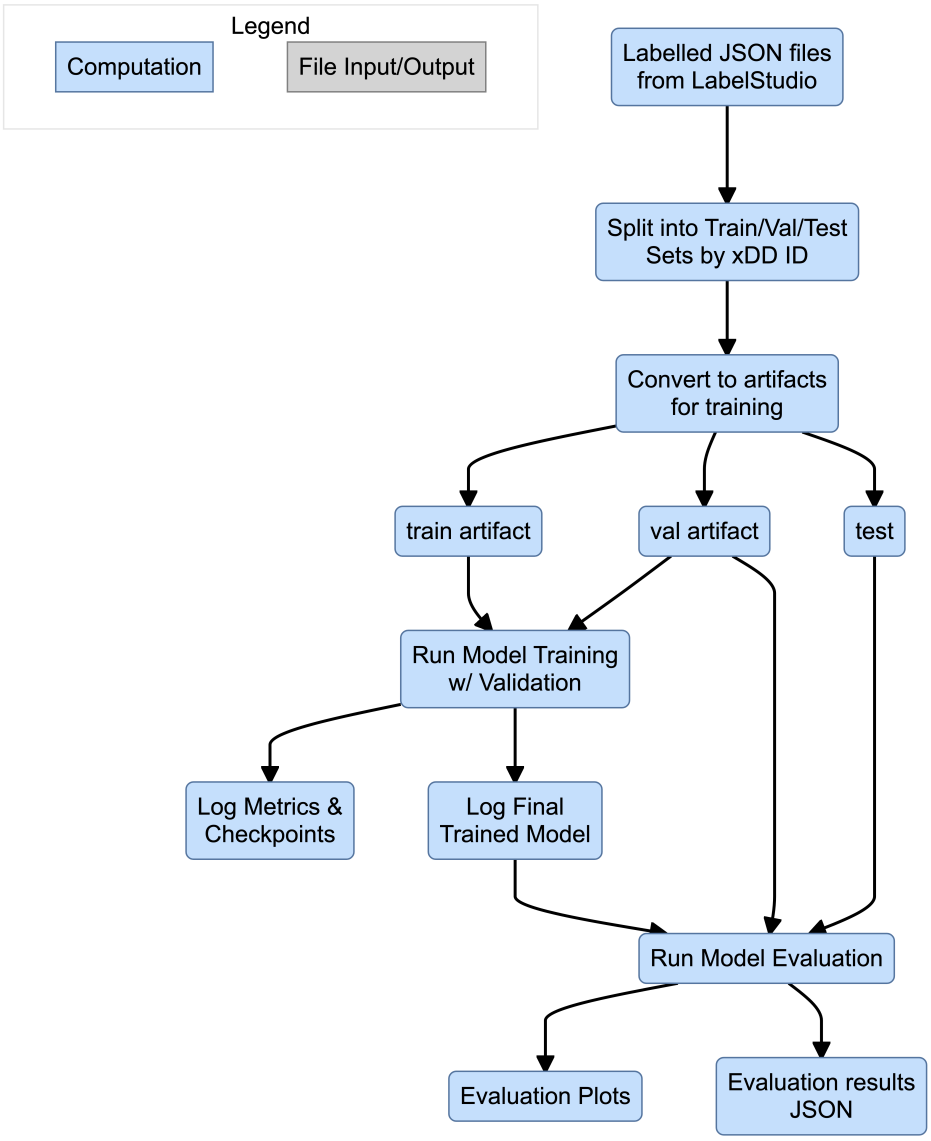
Table 9: spaCy CLI Final Hyperparameters

Parameters	**Notes**
Batch size	• Maximized to utilize all available GPU memory, 128 for transformer based model and 512 for word vector based model
Epochs	• Initial runs with 15 epochs, observed evaluation loss minima occurring in first 7-13 depending on learning rate
Learning Rate	• Initial learning rate of 5e-5
Learning Rate Scheduler	• Warmup for 250 steps followed by a linear learning rate scheduler which linearly decreases learning rate across epochs
Regularization	• L2 (lambda = 0.01) with weight decay
Optimizer	• Adam (beta1 = 0.9, beta2=0.999)
Early stopping	• 1600 steps

Interesting difference to note here are the architectures of the NER models. Both the models use a base model to tokenize the input sentence and generate word embeddings, but the HuggingFace NER model fine-tunes a token classification head to it, which consists of a set of linear layer and soft max, whereas, the spaCy NER model chunks and labels a sequence of inputs using an algorithm similar to transition-based dependency parsing using Stack-LSTMs.

The workflow in Figure 2 shows the primary steps for training with the Entity Extraction models with intermediate files and processes.

Figure 2: This is how the Entity Extraction model training process runs with intermediate files and processes.



2.2.4 Model Evaluation

The target of extracting an article's data is to make that data discoverable within Neotoma. The target for the data extraction is to maximize extracted data recall as the cost of a data reviewer deleting a piece of data is much lower than having to go and read the article to add missed entities. Similarly, token based recall is chosen as the target metric which measures partial matches of data extracted. Entity based recall which measures exact matches of all parts of a word is considered as it reduces data reviewers time to review articles but the emphasis is placed on token recall for evaluating models.

For more in depth analysis of results the following methods are used to evaluate each model.

Table 10: Model Evaluation Methods

Method	Description
Strict	Exact boundary of extracted string and entity type matches the annotation.
Exact	Exact boundary of extracted string matches but does not discriminate by correct entity label
Partial	A partial boundary overlap of the extracted string and does not discriminate by correct entity label
Type	Any overlap of the correct entity type is considered correct.

From the Message Understanding conference (Chinchor and Sundheim 1993) the following detailed metrics are used during evaluation for each of the above methods:

Table 11: Model Evaluation Metrics

Metric	Description
Correct (COR)	Both the labelled entity and predicted entity are the same
Incorrect (INC)	The labelled entity and predicted entity do not match
Partial (PAR)	The labelled entity and predicted entity are similar but not the same
Missing (MIS)	A labelled entity is not captured by the model
Spurius (SPU)	The model predicts an entity where there is none in the labelled text

2.3 Data Review Tool

In order to for the Neotoma data stewards to view the results of the Article Relevance Prediction and the Article Data Extraction, the final data product required was an interactive dashboard. The goal of this data product is to manually review the output of the entire natural language processing pipeline. The users are able to make corrections to the extracted entities by comparing the extracted entity to the sentence or opening the full-text article. In

addition, the user is able to delete incorrectly extracted entities or add additional entities that the extraction model missed. The output of the Data Review Tool is a JSON object that can then be used to retrain the Article Entity Extraction model and populate the Neotoma database. This will lead to more information sharing and better results in the future, which will decrease the time required by the data stewards while reviewing the extracted entities of an article.

2.3.1 Approach

The project's objective was to develop a customizable and freely available product with custom components. Considering this, the decision was made to prioritize open-source solutions such as R Shiny (Chang et al. 2023) and Plotly Dash ("Dash" 2023) over paid alternatives like Tableau or Power BI. The choice between R Shiny and Plotly Dash was influenced by the project's customization requirements. During discussions with Neotoma, the decision to implement either of these options did not have any impact. Ultimately, the team concluded that Plotly Dash in Python would be the preferred choice for the ongoing development of the project dashboard as the rest of the pipeline was to be written in Python.

2.3.2 Evaluation

In order to create an interactive tool that would be appropriate and efficient for the reviewers who are using this tool, the aim was to make it user-friendly and simple to use. To accomplish this success criteria, the requirements outlined in Table 12 were developed along with the targets to make these requirements successful for the data reviewers.

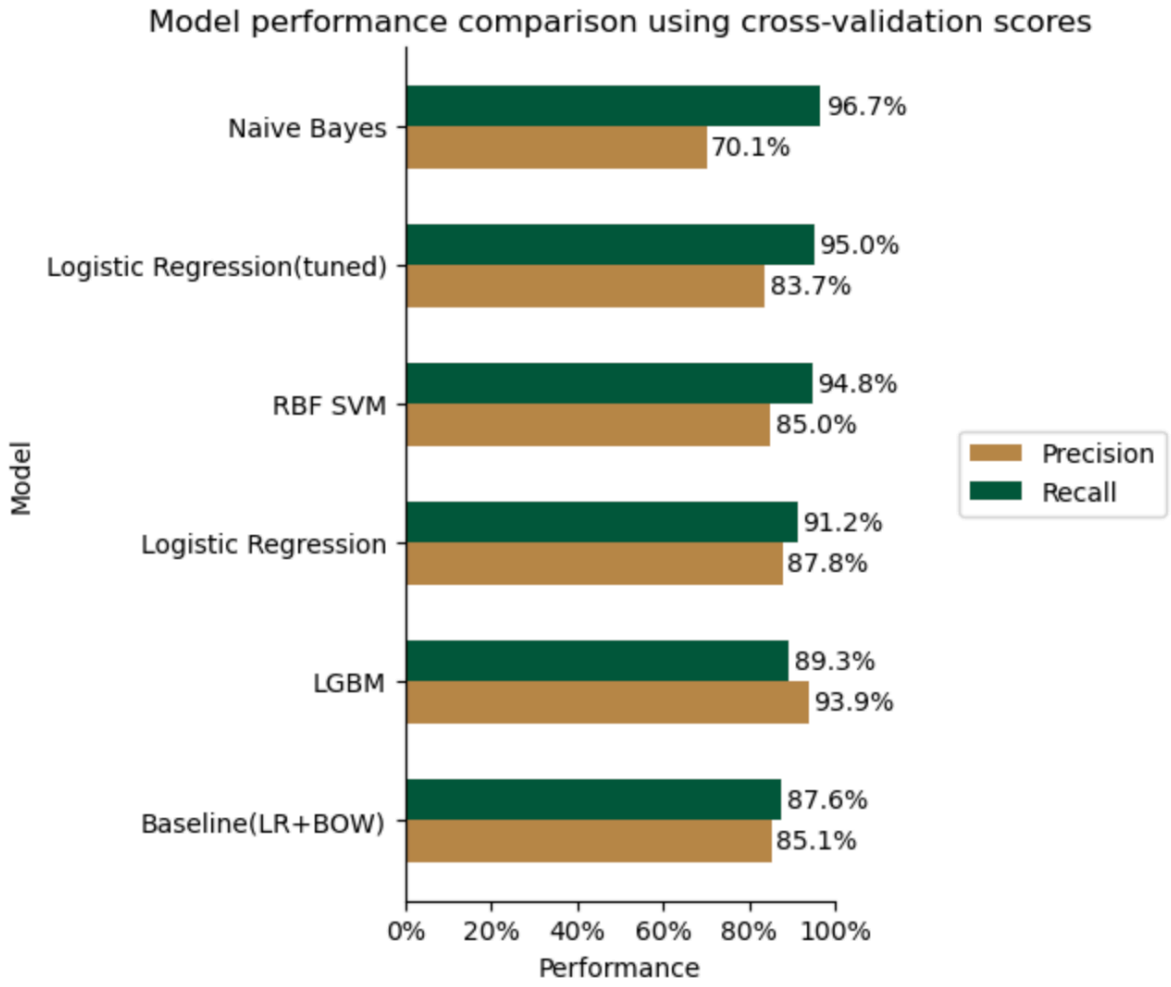
Table 12: Data Review Tool Target Metrics

Requirement	**Target**
Options for reviewing extracted data	Accept, Reject, Edit then Accept
Other data made available to the user	Article DOI, Journal Name, Hyperlink to Article
Displaying text from where the data was extracted	Current sentence and 1-2 sentences before/after.
User skill to run	Non-Technical (e.g. no code/CLI)
Number of mouse clicks to review single piece of data	1-2
Reviewing workflow	Able to save/resume progress.
Output file format	JSON

3 Data Products and Results

3.1 Article Relevance Prediction

Figure 3: Model Performance Comparison using Cross-Validation Scores



Among all the models trained, Naive Bayes has the highest recall score (96.7%), but its precision score was low (70.1%) and will introduce false positive. Among all gradient boosted models, LGBM has the highest precision (93.9%), but its recall score (89.3%) did not outperform logistic regression (94.8%). Among the analogy-based models, SVM with RBF kernel achieve a recall performance (94.8%) that's comparable to logistic regression (95.0%), and both are sufficiently high (Figure 3). The final decision was to use logistic regression with tuned hyperparameters for its good performance and better interpretability. On the test data set, the final model achieved a recall score of 96.5%, and a precision score of 85.2%. To put

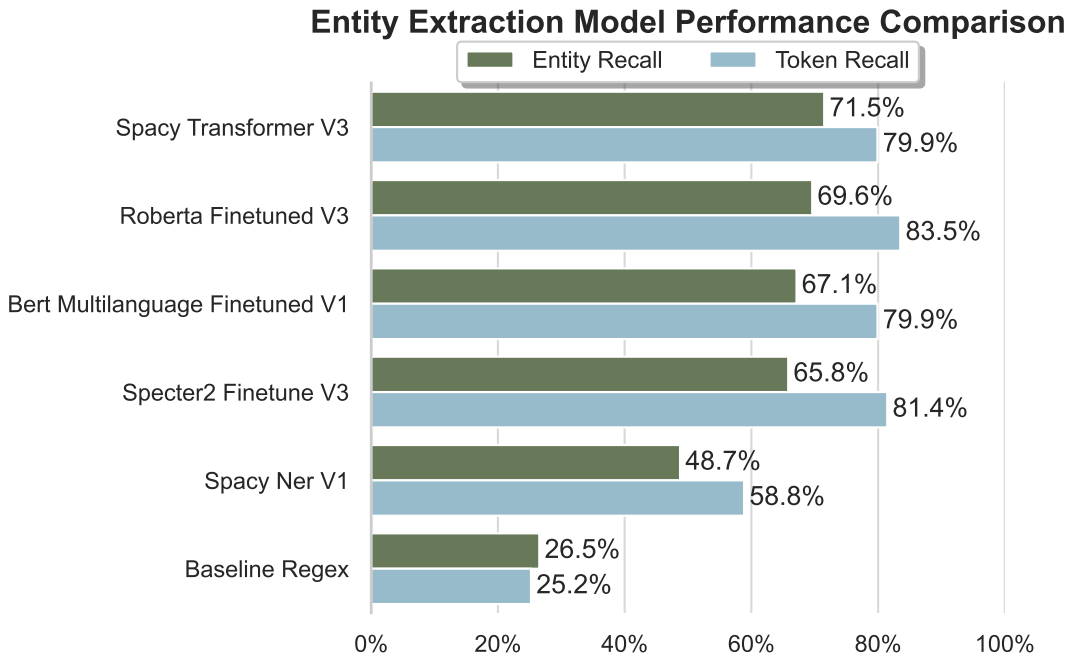
this into actual article counts, among the 666 articles in the test set, there are 5 false negatives and 24 false positives.

3.2 Article Data Extraction

Of the two approaches developed using the spaCy and Hugging Face model training systems there are two candidate models proposed to be used. The first is the Hugging Face finetuned RoBERTa based model which performed best of the Hugging Face models. The second is spaCy's transformer based model. The reason for proposing two models moving forward is the spaCy transformer model has better entity based recall along with better precision, meaning the extracted data is more often correctly extracted. Whereas the Hugging Face RoBERTa model achieves better token recall but has lower precision, meaning it detects more of the overall entities but contains more false detections which must be deleted by the reviewers. The slight differences in the accuracies and varying strengths can be attributed to the different NER models leveraging the RoBERTa-base transformer embeddings differently.

Figure 4 shows the primary candidate models token and entity based recall.

Figure 4: The entity and token based recall for each model on the test set.



The token based recall for each model on the test set by entity type are shown in Table 13.

Table 13: Entity extraction model results for token based recall, overall and by entity type

Model	Recall (%)	Age (%)	Site (%)	Taxa (%)	Region (%)	Geog (%)	Email (%)	Alti (%)
RoBERTa	83.5	82.4	87.3	90.4	73.7	77.6	38.1	84
Specter2	81.4	86.7	83	86.9	71.8	62.1	47.6	53.1
BERT Multi-lang.	79.9	81.1	76.9	85.5	74.9	58.6	33.3	81.5
SpaCy Transf.	79.9	82.2	73.7	88	70.6	64.7	100	83.3
SpaCy NER	58.8	63.6	23	69.7	66.2	35.3	0	26.2
Baseline Regex	25.2	9	0	46.3	26.9	0	100	27.2

An important observation to make here is that even the top models have a lower precision score for the SITE names and REGION names. The models gets confused when deciding whether an entity should be classified as a SITE or a REGION. This is partially due to quality of labeling entities as well as the fact that both these types correspond to the name of a place or a wider area. See Figure 5 for a confusion matrix generated using the test set assets highlights the issue.

3.3 Data Review Tool

The final data review tool that was created was a multi-page Plotly Dash application. The tool can be replicated by launching Docker containers, enabling anyone within the Neotoma community to easily launch and utilize the tool for reviewing outputs from the entity extraction pipeline. This approach ensures accessibility and convenience for users who wish to review the pipeline outputs within the Neotoma project.

This web application enables users to review the extracted entities from articles, make changes, add additional missed entities and remove inaccurately extracted entities. The application includes a button to launch the article in an external browser tab so that the reviewer can verify beyond the current sentence and sentence preceding/following that was provided as output. The entire review process does not require any coding knowledge, and reviewers can navigate through the review workflow in three clicks or fewer from the Article Review page Figure 6.

The output of this data review tool is a JSON object that stores the originally extracted entities as well as the corrected entities. The Neotoma organization can utilize these entities to add new paleoecology articles to the database. In addition, the reviewed entities can be fed back

Figure 5: Confusion Matrix for RoBERTa Model

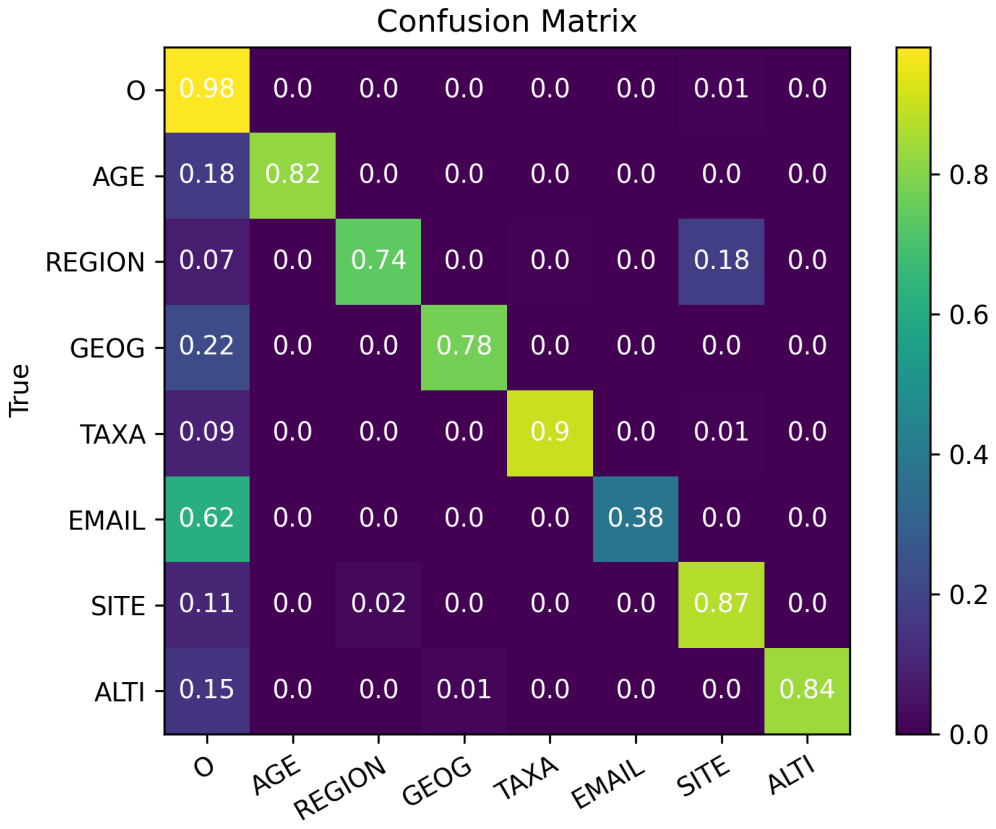
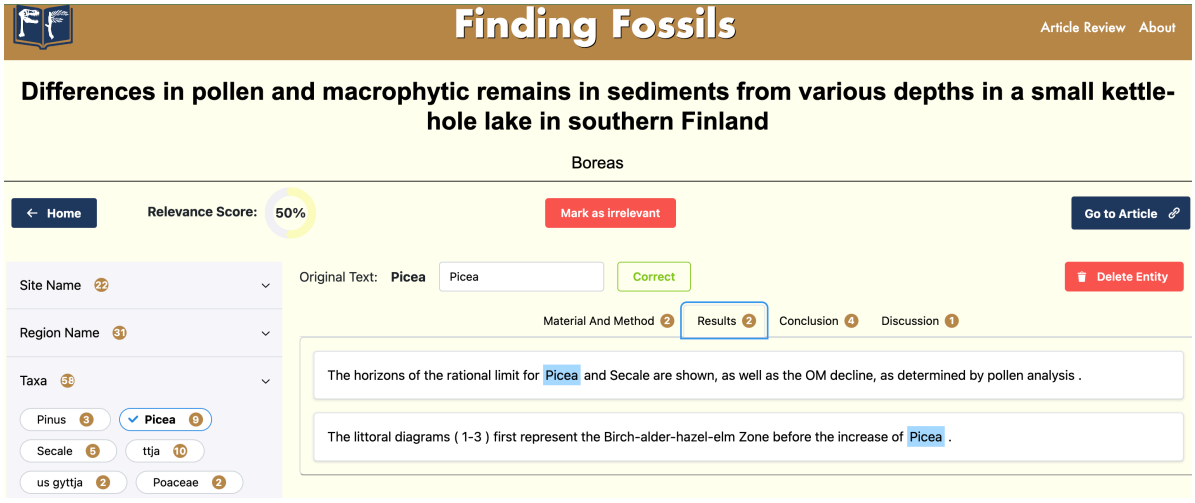


Figure 6: Data Review Tool



to model using our model retraining pipeline, contributing to the continuous improvement of the quality of extraction of entities.

Table 14: Data Review Tool Mertic Results

Requirement	**Results**
Options for reviewing extracted data	Accept, Reject, Edit, Add then Accept
Other data made available to the user	Article DOI, Hyperlink to Article
Displaying text from where the data was extracted	Current sentence and 1 sentence before/after.
User skill to run	Non-Technical (e.g. no code/CLI)
Number of mouse clicks to review a single piece of data	3 clicks from launch
Reviewing workflow	Able to save/resume progress.
Output file format	JSON

As Table 14 shows, all of the requirements that were originally laid out in the proposal were implemented through this Dash application. However, there are further improvements that could be made if there was more time available. One crucial improvement would involve incorporating an additional page within the dashboard to enable users to review the results of the Article Relevance Prediction model. This page would present both positive predictions (articles relevant to Neotoma) and negative predictions (articles not relevant). By allowing Neotoma data stewards to review and correct these predictions, the accuracy of the Article Relevance model can be improved through retraining. At present, the data review tool includes a button in which the user can mark an article as irrelevant. This can then be used to retrain the model by labelling it as not relevant to Neotoma. However, there is no current method implemented to make corrections on negative predictions. If this was addressed in the future, the original bias from the sample of data that was used to train the Article relevance tool could be minimized which would lead to greater model performance.

3.4 Product Deployment

The end goal of this project is to have each data product running in a semi/un-supervised fashion. The article relevance prediction pipeline is containerized using Docker. It is expected to run on a daily daily or weekly basis by the Neotoma and gets the latest published articles from the public xDD API, runs the article relevance prediction, and finally submits relevant articles to xDD to have their full text processed.

The Article Data Extraction pipeline is containerized using Docker and contains the entity extraction model within it. It is run on the xDD servers as xDD is not legally allowed to send full text articles off their servers. The container accepts full text articles, extracts the entities,

and outputs a single JSON object for each article which is then exported by xDD back to the Neotoma team. The extracted entity JSON objects are combined with the article relevance prediction results and this is what is loaded by the Data Review Tool. Figure 7 depicts the work flow.

4 Conclusion and Recommendations

4.1 Conclusions

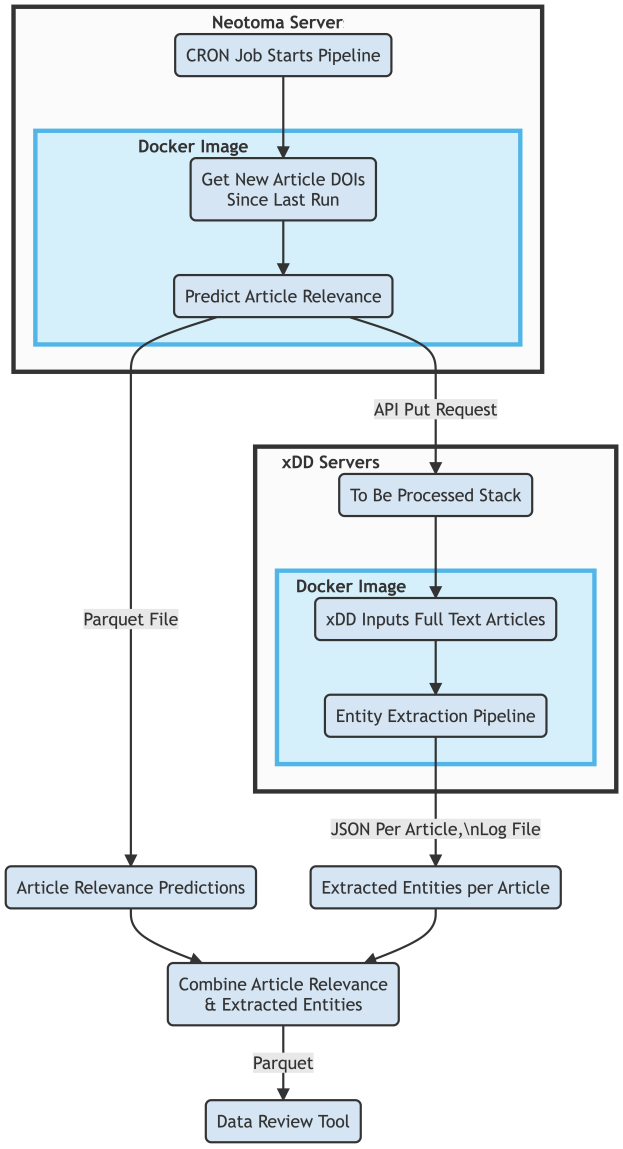
The Neotoma database plays a crucial role in paleoecological research by providing data on ecological changes over the past 5 million years. However, the reliance on manual submissions by researchers has led to difficulties in data entry and possibly hindered collaborative efforts to fully understand these ecological changes. The implementation of our pipeline brings significant benefits to the Neotoma community and paleoecological research. Firstly, by automating the search for submissions of new research articles that are relevant to the Neotoma database through the Article Relevance Prediction this process will growth of the Neotoma community, attracting more researchers and fostering collaboration in ecological research. Secondly, researchers will experience reduced time and effort required for data uploads, as the manual submission process is replaced with automated data extraction and entry through both the Article Entity Extraction and Data Review Tool. Finally, with the addition of access to the data, researchers in the Neotoma community may be able to answer questions about ecological changes that were previously unknown

4.2 Recommendations

Due to time constraints, improvements were identified but could not be implemented within the project's timeframe. To provide a concise summary for future project contributors, three essential features have been identified that would significantly enhance the overall pipeline.

1. **Relevance Review Page:** incorporating an article would help minimize the bias present in the original model and improve the overall accuracy. This additional step would allow for a thorough assessment of article relevance, reducing both false positives and false negatives in the predictions. It will also reduce the bias in the training dataset that currently exists due to our data collection strategy.
2. **Adding new entities:** Although the focus was primarily on accurately extracting the required entities, there are additional entities that are considered beneficial but not mandatory by the Neotoma team. To address this, the Label Studio setup that was created through the project enables future contributors to easily add these additional entities. By implementing these changes, the pipeline can be further refined and optimized to benefit the Neotoma community at large.

Figure 7: This is how the MetaExtractor pipeline flows between the different components.



3. There are likely many corrections that will be performed frequently on the data review tool by the Neotoma Data Stewards. Therefore, a model could be implemented to learn these corrections and preprocess these corrections in future improving efficiency and reducing time spent reviewing.
4. **Computer Vision:** Using PDF images and OCR technology to extract figures and table data for richer information extraction and inclusion of higher quality data into the Neotoma database.

5 Acknowledgements

Data were obtained from the Neotoma Paleoecology Database (http://www.neotomadb.org) and its constituent database(s). The work of data contributors, data stewards, and the Neotoma community is gratefully acknowledged.

A huge thanks to Simon Goring & Socorro Dominguez from the Neotoma Database team for their support on this project.

References

AI, Borealis. 2023. "Tutorial 17: Transformers III - Training." https://www.borealisai.com/research-blogs/tutorial-17-transformers-iii-training/.

Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2023. *Shiny: Web Application Framework for r*. https://shiny.rstudio.com/.

Chinchor, Nancy, and Beth Sundheim. 1993. "MUC-5 Evaluation Metrics." In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*. https://aclanthology.org/M93-1007.

Crossref. 2023. "Crossref REST API." https://www.crossref.org/services/metadata-delivery/rest-api/.

"Dash." 2023. Plotly. https://dash.plotly.com.

Williams, J. W., E. G. Grimm, J. Blois, D. F. Charles, E. Davis, S. J. Goring, R. Graham, et al. 2018. "The Neotoma Paleoecology Database: A Multi-Proxy, International Community-Curated Data Resource." *Quaternary Research* 89: 156–77.