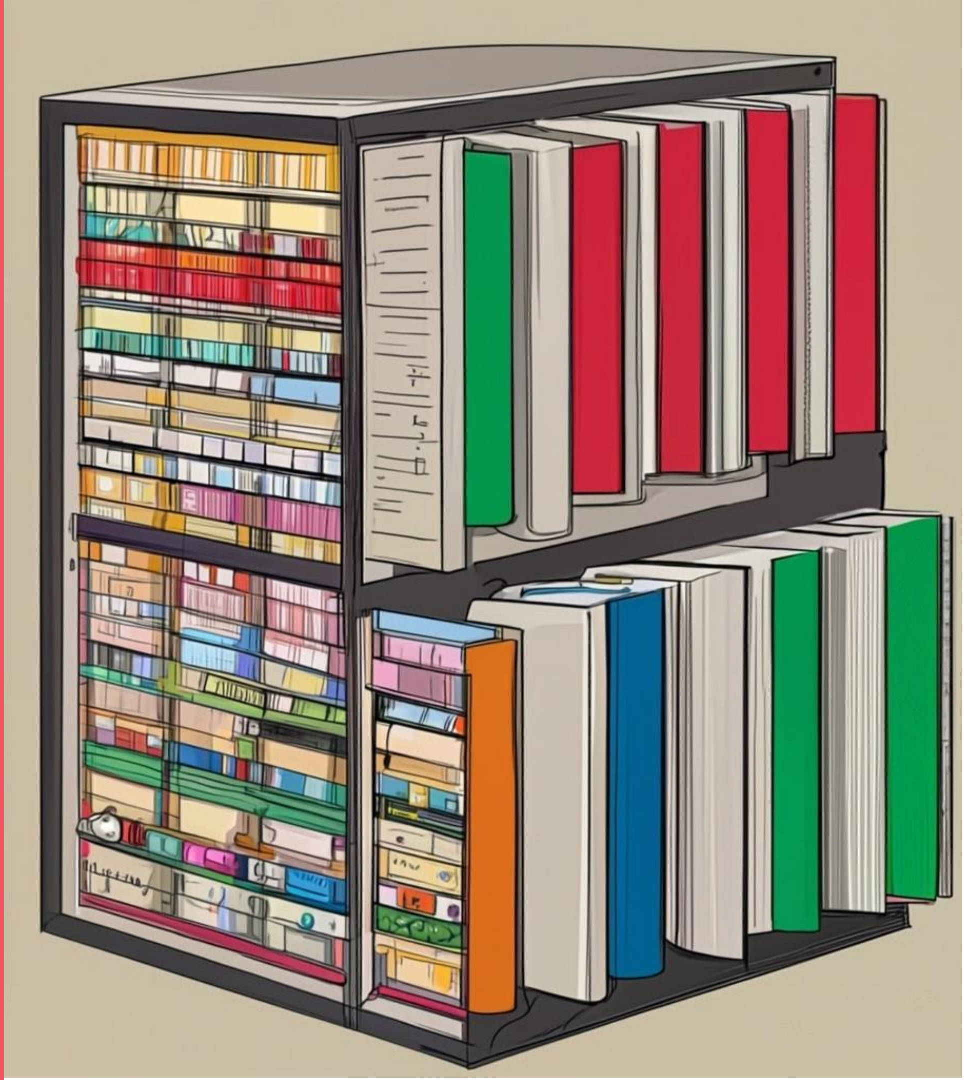
Library of Congress Labs

Exploring Computational Description

Final Presentation

Background

Introduction



LC Labs, a division of the Library of Congress (Library) Digital Strategy Directorate (DSD) in the Office of the Chief Information Officer (OCIO), leads a **program of experimentation** that includes user-centered research, prototyping and development of emerging methods, workflows and functionalities that **connect** Library collections, data, services and expertise **to users in new ways**.

Exploring Computational Description was an experiment designed to answer a number of research questions.

Research Questions

What are examples, benefits, risks, costs and quality benchmarks of **automated methods for creating workflows to generate cataloging metadata** for large sets of Library of Congress digital materials?

What **technologies and workflow models** are most promising to support metadata creation and assist with cataloging workflows?

What **similar activities** are being employed by other organizations?

Background

Requirements

Test and report on at least five (5) machine learning models or methods to detect or generate full level bibliographic records whenever possible from the textual and/or visual elements of ebooks in epub, PDF or other digital formats.

The minimum fields to be generated are: titles, author names, unique identifiers, date of issuance, date of creation, genre/form and subject terms.

Approximately 20,000 existing MARC records and ebooks made available by the Library for training data.

Explore and experiment with at least two (2) more machine learning techniques that could augment cataloging workflows at the Library and enhance access to Library collections.

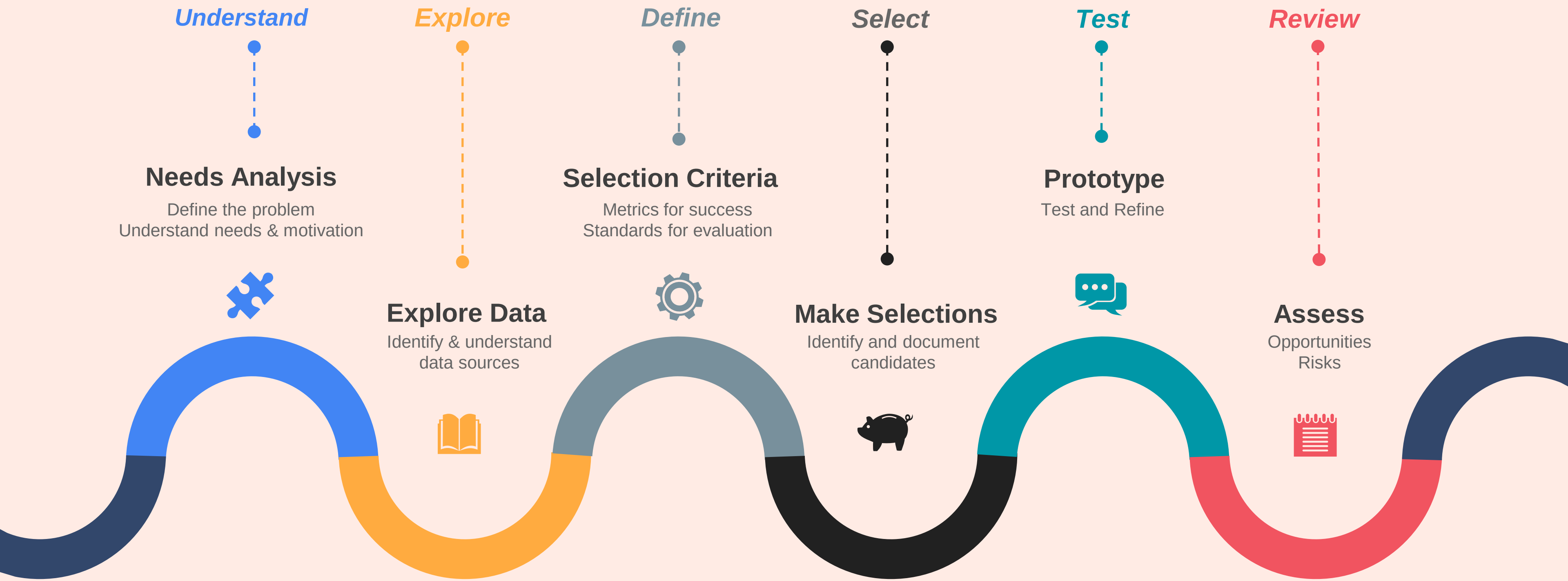
- How are other institutions making use of automated methods to assist in the generation of bibliographic data or catalog records?
- What are the potential benefits, costs and risks of using such methods?
- How quantifiably good are the current state of the art methods at generating catalog data?
- Are there particular areas where automated methods work well?
- Areas where they work badly?
- Are there certain types of metadata where the Library could productively explore automated cataloging?
- Are there certain types of metadata where the current state of the art just isn’t good enough for use in a production cataloging workflow?
- Of the currently available technologies and approaches which are the most promising?
- What are the strengths and weaknesses of each?
- How can the library assess these workflows?
- What might be the next steps in iterating towards a production ready workflow?

Introduction

Process

Experiment Process

Overview



Understand

Needs Analysis

Define the problem
Understand needs & motivation

Explore

Explore Data

Identify & understand
data sources

Define

Selection Criteria

Metrics for success
Standards for evaluation

Select

Make Selections

Identify and document
candidates

Test

Prototype

Test and Refine

Review

Assess

Opportunities
Risks

Define the problem

- What are the priorities of the Library as an institution?
- Who are the users?
- What are their needs and motivations?
- What are the challenges for the Library?
- What are the challenges for users?

What data is available?

- Formats
- Statistical properties
- Relevant features
- Challenges with the data
- How balanced / unbalanced is the data?

Where are we?

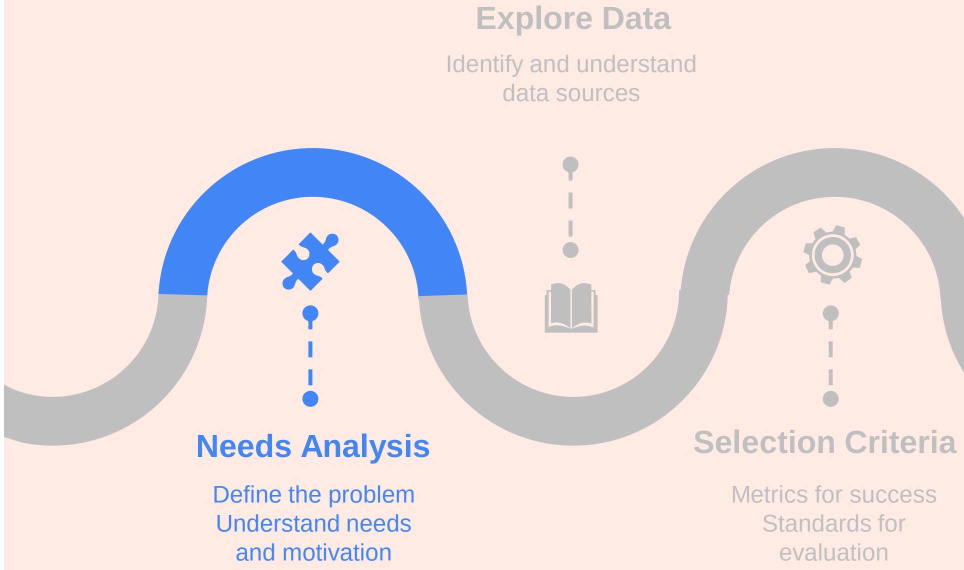
- Landscape analysis
- Define quantitative criteria and metrics for assessment
- Identify qualitative criteria for assessment

- Assess candidates from landscape analysis:
    - Selection criteria
    - Diversity of Approach
- Select five for prototyping

- Create:
  - Training data
  - Test and validation data
- Measure against selected metrics
- Train or fine-tune
- Repeat

- Evaluate against
  - Institutional requirements
  - User needs
- What are:
  - Benefits
  - Risks
  - Costs
  - Expected performance / benchmarks
- Which are the most promising approaches?

Needs Analysis



Explore Data

Identify and understand data sources

Needs Analysis

Define the problem
Understand needs and motivation

Selection Criteria

Metrics for success
Standards for evaluation

This experiment was primarily focused on **evaluating and testing approaches to automated generation of catalog metadata**, rather than on a deeper exploration of the needs of users.

Initial workshops were used to identify institutional priorities, and to develop a high-level understanding of the needs of catalogers.

A follow-on project *Towards Piloting Computational Description* has a deeper focus on understanding and analyzing user needs.

Institutional Priorities

In an initial workshop key stakeholders were asked to review the potential deliverables from the experiment and vote in order to indicate which of the potential outputs was a priority.

- **Testable assisted cataloging workflows** providing support for catalogers using computational descriptions
- **Clear metrics** for measuring the accuracy and quality of computational descriptions (in general)
- **Performance data** for the models tested (specifically)
- **General comparison** / documentation for the models tested (specifically)
- **Assessment** of risks, benefits and costs to the Library

We also asked stakeholders to vote on which fields would be most useful, with the highest votes for:

- **Unique identifiers**
- **Subject terms**
- **Author** (or other personal names)

User Needs

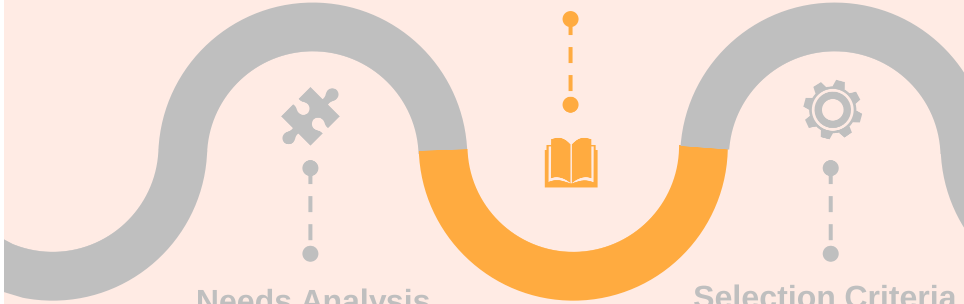
- Users, in this instance, are understood as expert catalogers tasked with cataloging e-books.
  - Focus was not in end users of the library catalog
  - Focus was not on a deep dive into the end to end cataloging workflow
- Workshops identified a number of key pain points:
  - **Subject / genre cataloging**:
    - Choosing the right heading
    - Time involved in creating relevant strings
    - Scope notes may not provide enough information to clarify applicability
  - **Personal Names**:
    - Determination of the appropriate authority record
    - Disambiguation between multiple potential matches
    - Different forms of the same name
  - **E-books**:
    - Not necessarily easy to process through current workflows

What we learned

The needs and priorities identified through the workshops:

1. Informed our selection criteria and methods of evaluation
2. Placed clear metrics for measuring accuracy and quality at the heart of the process
3. Helped us ensure that the most important fields were in scope for the training and evaluation:
    a. Unique identifiers
    b. Subject terms
    c. Personal Names
4. Provided clear direction that the assisted cataloging prototypes should be targeted at:
    a. Subject classification
    b. Personal Names and Authority Control

Explore Data



Explore Data

Identify and understand
data sources

Needs Analysis

Define the problem
Understand needs
and motivation

Selection Criteria

Metrics for success
Standards for
evaluation

Exploring the data

The data provided for this experiment consisted of:

- 23,000 ebooks in PDF and EPUB format
- MARCXML records for each ebook

The e-books were primarily in English, with small numbers in other languages.

The e-books were from four collections:

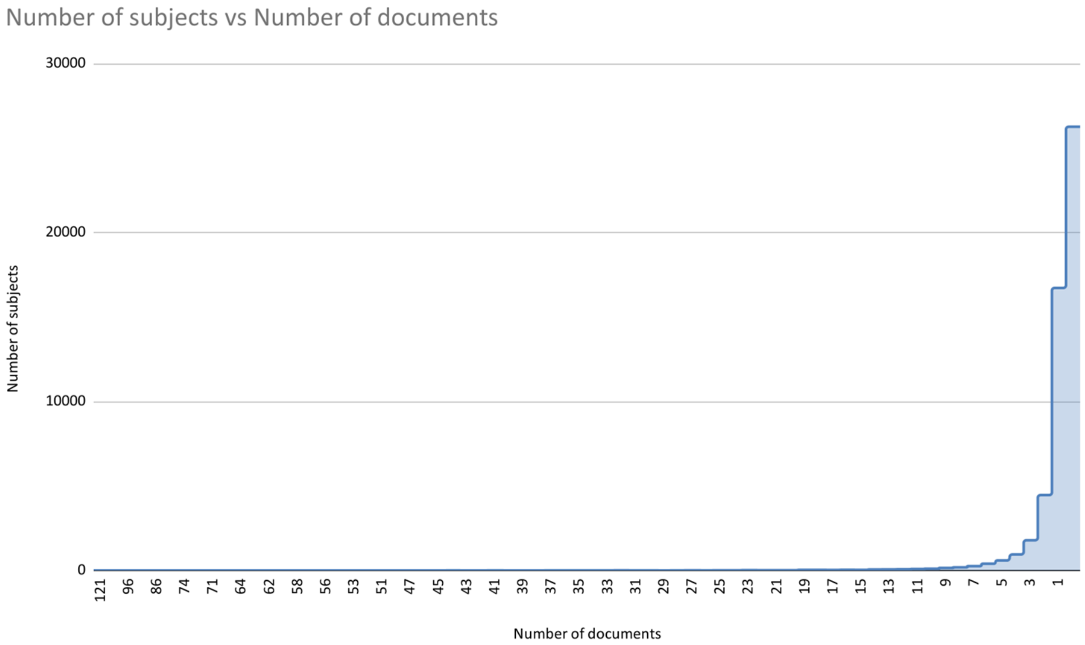
- Open Access E-books
- Legal Reports
- E-Deposit Registration E-books
- Cataloging-In-Publication E-books

Subjects

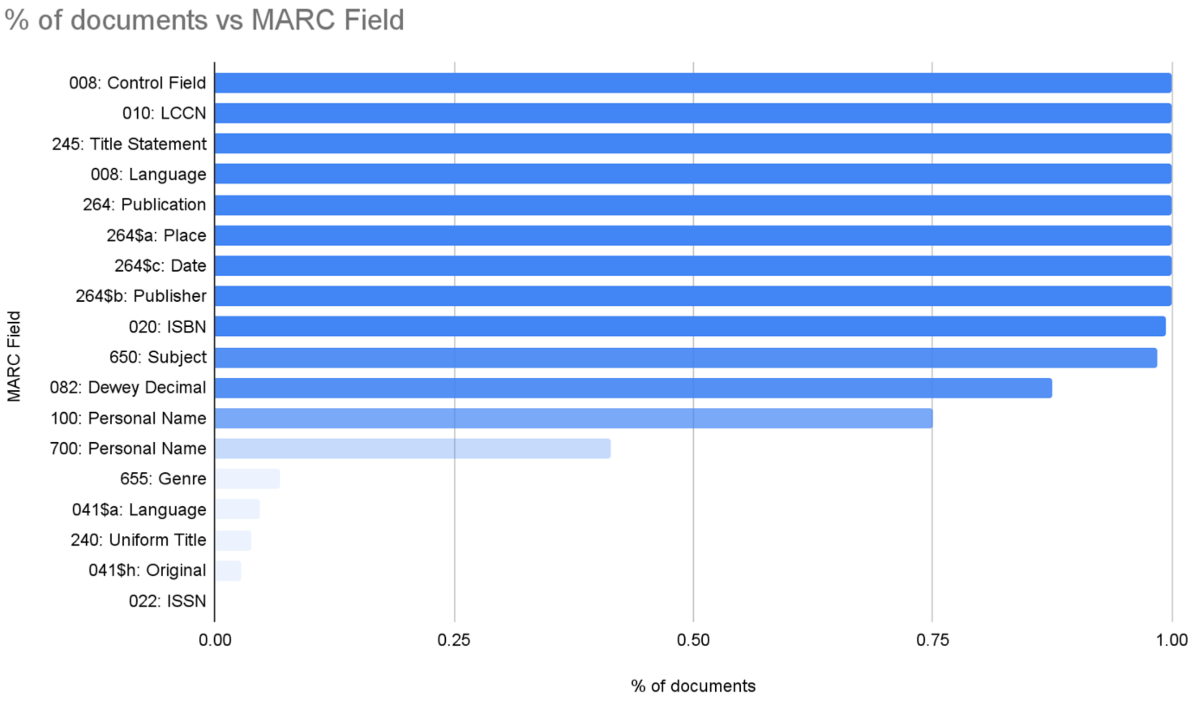
Across the corpus of 23,000 e-books there were approximately:

- 26,000 different LCSH subject headings used
- Only approximately 1,800 LCSH subject headings appeared more 3 or more time

The subjects form a very *unbalanced* dataset, with a long tail of subjects appearing only once, and a very small number of subjects appearing often.



Distribution of LCSH subject terms by frequency of occurrence



Distribution of MARC fields across the document corpus

What we learned

Based on the project requirements and the most commonly occurring fields, we identified the following fields as candidates for testing for automated metadata generation

- 010: Library of Congress Control Number (LCCN)
- 020: International Standard Book Number (ISBN)
- 245: Title Statement
- 264: Production, Publication, Distribution, Manufacture, and Copyright Notice
- 650: Subject Added Entry - Topical Term
- 100: Main Entry - Personal Name
- 700: Added Entry - Personal Name

With multiple subfields used for training and evaluation on MARC 100, 700, 245 and 264.

Field choices were influenced by:

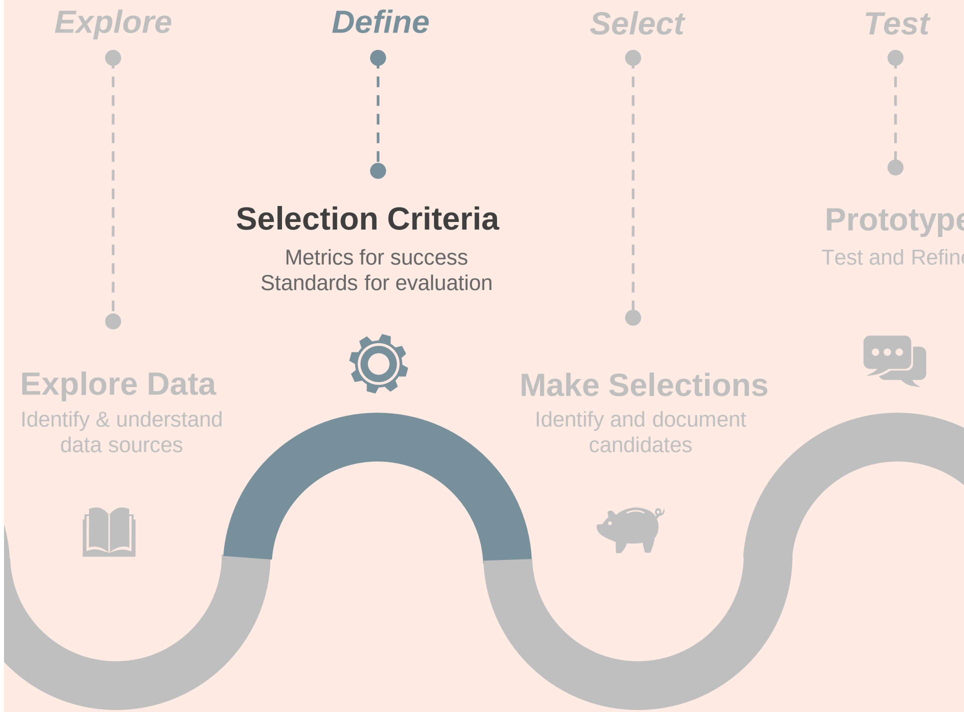
- Priorities from the initial stakeholder workshop
- Pain points identified during the needs analysis session with catalogers

What we learned

We also identified that Subject Classification was likely to be challenging

- The number of **subjects** to number of documents is high
- The number of **instances** of each individual subject are very low, with most subjects only appearing once in the entire corpus
- A very small number of **subjects** appear many times
- Subjects, as a whole, are very unbalanced across the entire dataset

Selection Criteria



The potential array of machine learning tools, models, and workflows that can be applied to e-book texts is vast. Hundreds of tools are launched every year and **the academic literature is full of promising approaches for generating useful information from text**.

In order to select a practical, useful, and informative spread of approaches for the five (or seven) prototypes for this experiment, **we identified a number of key selection criteria**.

What are we selecting?

When selecting approaches to prototype as part of the experiment. We are talking about selecting an entire end-to-end pipeline or workflow that takes text (or other data) as input and produces metadata as output that can be transformed into MARC (or BIBFRAME) for reuse.

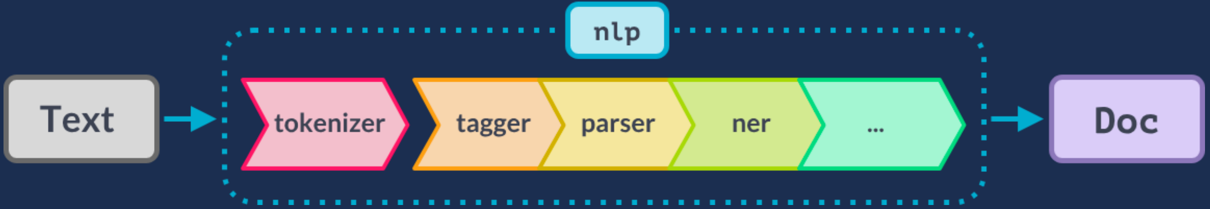


Diagram from https://spacy.io/usage/spacy-101

Library / Framework

Natural language processing libraries like Spacy, NLTK, Hugging Face Transformers, or StanfordNLP typically can carry out many different tasks:

- Tokenizing
- Parts of Speech
- Entity Recognition / Token Classification
- Text Classification
- Summarization
- Keyword Extraction

Libraries of this type also typically provide tooling for:

- Training or fine-tuning models
- Evaluation
- Packaging

Many libraries support multiple models.

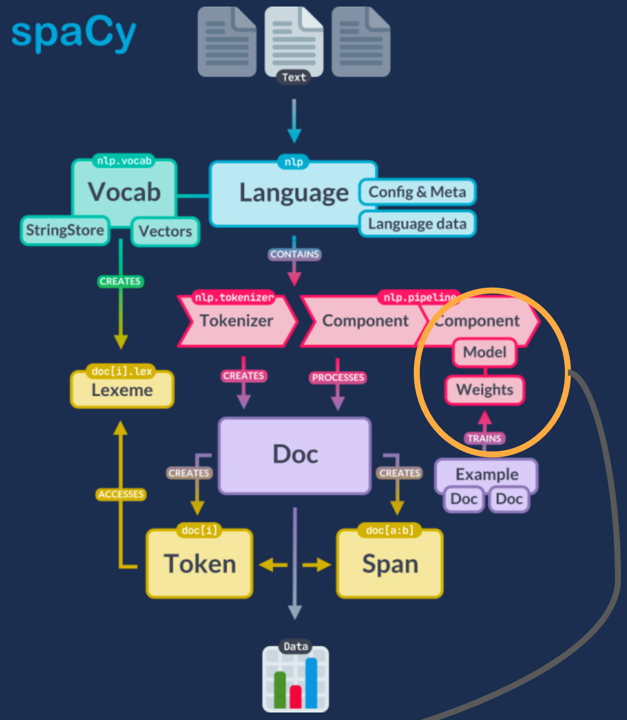
Diagram from https://spacy.io/usage/spacy-101

Model

A **machine learning model** is a mathematical representation of a real-world process, trained using data, that makes **predictions or decisions based on new input data**.

Typically a **model** is a component in a larger workflow, although often the term model is used to refer to the complete end-to-end workflow.

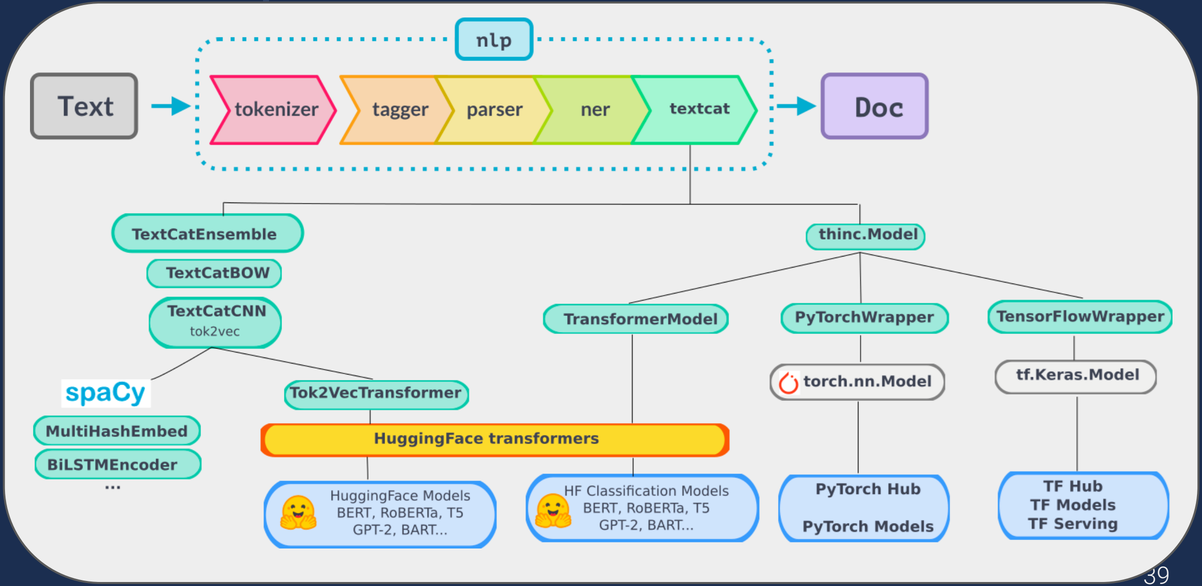
The example on the right shows **the architecture of a Spacy text classification pipeline**. Note that the language model is one component in this architecture.



Model

Often, when talking about models, we are referring to a **pre-trained model which has been trained on existing data** and which can then be used to make predictions on new data.

Many libraries support the download and reuse of existing models from hubs such as *Hugging Face*, or provide a suite of pretrained models which can be used as is or fine-tuned on specific data.



Architecture

Models are built on top of an architecture which defines how the model **accepts input**, how the model **is trained**, and how the model **produces output data**.

The architecture alone is **not** a *model* and the same architecture, such as the *Transformer* architecture may be the basis for hundreds of different models and/or libraries.

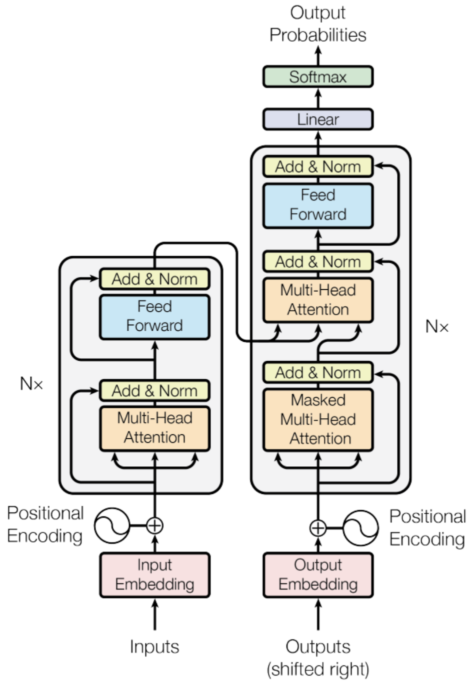


Figure 1: The Transformer - model architecture.

There is not a one-to-one prototype to model relationship

Each of our **prototypes** uses a different library or framework.

Several prototypes test more than one **model** and more than one **architecture** because the library of framework provides multiple pipeline components and multiple model architectures.

The same model can potentially be used by different libraries or frameworks.

Quantitative

Performance and metrics

It seems obvious that the prototypes that form part of this experiment should test the *best* tools for job.

When selecting machine learning / AI tools to evaluate, we ought to select the highest performing tools for evaluation.

However, it is not always immediately clear which tools really are the best for the job, or how we might identify them.

Select the right metrics

There are many metrics that are used to evaluate machine learning models and workflows.

- Accuracy
- Precision
- Recall
- F1 score
- ROUGE
- METEOR
- NDCG (normalized discounted cumulative gain)
- etc

Selecting the right metric for comparison between models during the prototyping and for ranking models during the selection phase—based on published data—is important.

Generally, we used F1-Score as the preferred metrics for token classification tasks, and for text classification tasks NDCG or F1 @5 or @3 (that is, the metrics when computed only over the top ranked 5 or top ranked 3 results).

Identify comparable datasets

Performance on published metrics is a useful guide to how successful a model might be on LIbrary e-book data, but is likely to be a more accurate guide when the datasets on which a model has been evaluated for published metrics are relevantly similar to the Library e-book data.

For example, many published text classification metrics are based on:

- Very short texts
- Very small number of classes, usually well under 100 classes

Lists of datasets at sites such as:

Machine Learning Datasets | Papers With Code

Find Open Datasets and Machine Learning Projects | Kaggle

Hugging Face – The AI community building the future.

Can provide a guide to ensure that leaderboards and comparative metrics are relevant to the Library experiment.

Review published performance

A great many published leaderboards of performance exist for common natural language processing tasks such as summarization, machine translation, question answering, text classification, and entity recognition or token classification.

Our approach was to review the published performance of candidate models against published benchmarks on relevantly similar data at:

The Extreme Classification Repository: Multi-label Datasets and Code

Browse the State-of-the-Art in Machine Learning | Papers With Code

NLP-progress

And others.

In addition, there are many review articles and published papers assessing the performance of a range of models on standard published datasets and we identified promising models through literature review.

Qualitative

Practicality and ease of evaluation

Pragmatic Evaluation

- ***Diversity of approach***: all things being equal, we preferred to include a diverse range of approaches in our prototypes to ensure we cast as wide a net as possible rather than testing narrowly similar approaches
- ***Developer friendliness***: is the model/workflow/tool easy to work with? Can developers easily embed the tool in a Library workflow? How easy is it to customize or configure? Is the code written in a widely supported language? Does it make use of industry standard best practice?
- ***Documentation***: how well documented is this tool, framework or model? Is the documentation up to date? Is the documentation clear and well-written? Does the documentation cover the specific uses required for the experiments?

Pragmatic Evaluation

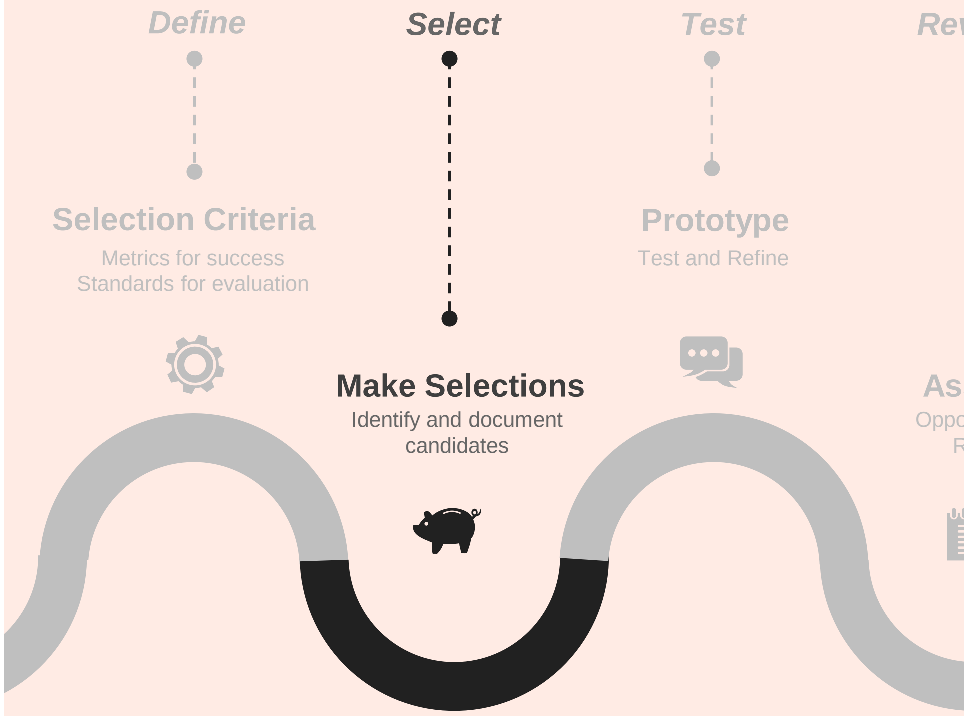
- ***Project activity and responsiveness to issues***: is the library/tool/model under active development? How recent was the latest update? Are the developers responding to issues and bugs raised?
- ***Ease of generating training data***: Does the model use standard data formats? How easy is it to generate training data at scale from LoC records?
- ***Reliability:*** How reliable was the model during our preliminary review?
- ***Compute cost***: How resource hungry was the model during our preliminary review? What might the projected costs be?

The experiment had limited time and a limited budget for evaluating approaches to computational description so practical concerns were also important.

In order to select approaches that were likely to produce good outcomes, we used a mixture of desk research and practical assessment where we tested multiple approaches in an initial pre-selection exploratory phase.

All things being equally, we preferred to select practically testable models where the performance was similar.

Selection



Define

Selection Criteria

Metrics for success
Standards for evaluation

Select

Make Selections

Identify and document
candidates

Test

Prototype

Test and Refine

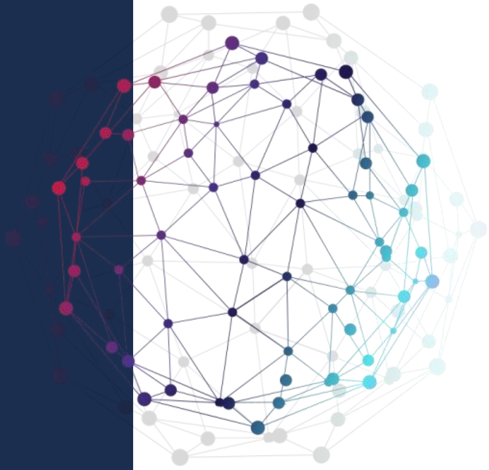
Rev

As

Oppo
R

Selection

Computational Description as a Machine Learning Problem



The tasks for *Exploring Computational Description* can be understood as instances of two common problems in natural language processing (NLP):

- **Token classification**. Also known as sequence classification, or sometimes text extraction or entity recognition.
- **Text classification**.

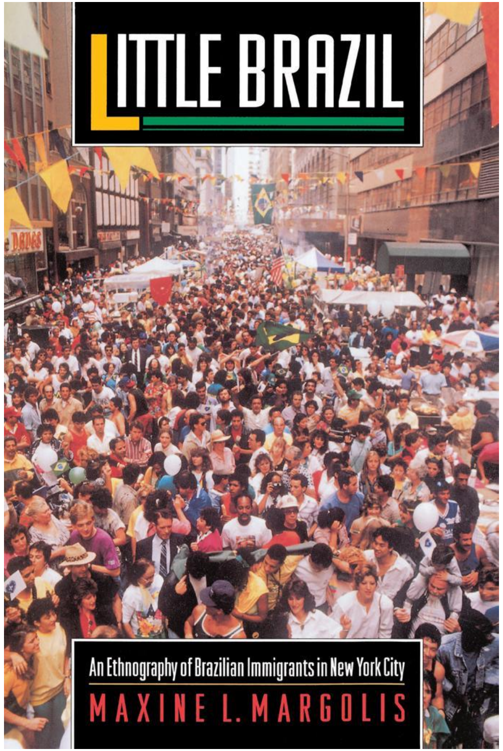
Both of these are instances of *supervised learning* in which existing labeled data is used to train or fine-tune machine learning workflows.

Token Classification

Token classification is the process of identifying groups of tokens—usually words, or parts of words—in a text and assigning them to particular classes or categories.

Or, for a given category or class, returning all of the groups of tokens that fall under that category or class.

For example, we want our machine learning model to be able to identify when a group of words (or tokens) is the name of the author of a work, or a title statement, or the date of publication.



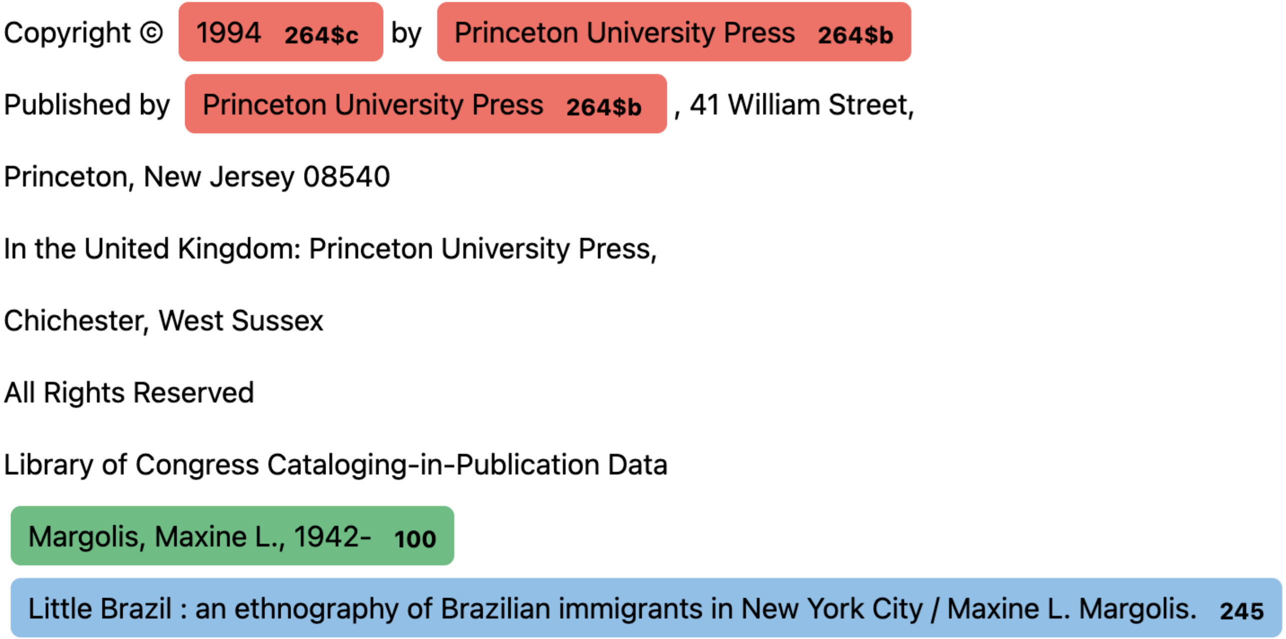
Published by Princeton University Press, 41 William Street,
Princeton, New Jersey 08540
In the United Kingdom: Princeton University Press,
Chichester, West Sussex

Library of Congress Cataloging-in-Publication Data

Margolis, Maxine L., 1942-
Little Brazil : an ethnography of Brazilian immigrants in New York City /
Maxine L. Margolis.
p. cm.
Includes bibliographical references and index.
ISBN 0-691-03348-X (cl.)
ISBN 0-691-00056-5 (pbk.)
eISBN 978-1-40085-175-1 (ebook)
1. Brazilian Americans—New York (N.Y.)—Social life and
customs. 2. New York (N.Y.)—Social life and customs.
I. Title.
F128.9.B68M37 1993
974.7'1004698—dc20 93-13699 CIP

R0

Token Classification



Copyright © 1994 264$c by Princeton University Press 264$b

Published by Princeton University Press 264$b , 41 William Street,

Princeton, New Jersey 08540

In the United Kingdom: Princeton University Press,

Chichester, West Sussex

All Rights Reserved

Library of Congress Cataloging-in-Publication Data

Margolis, Maxine L., 1942- 100

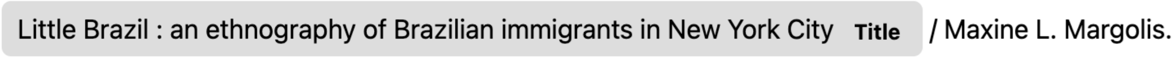
Little Brazil : an ethnography of Brazilian immigrants in New York City / Maxine L. Margolis. 245

Token Classification

Given a group of tokens (or words):

```
0: 'Little', 1: 'Brazil', 2: ':', 3: 'an', 4: 'ethnography', 5: 'of', 6:
'Brazilian', 7: 'immigrants', 8: 'in', 9: 'New', 10: 'York', 11: 'City',
12: '/', 13: 'Maxine', 14: 'L.', 15: 'Margolis.'
```

We want our machine learning model to successfully identify that tokens 0 through 12 correspond to the Title of the work,



Little Brazil : an ethnography of Brazilian immigrants in New York City **Title** / Maxine L. Margolis.

and ideally also that tokens 13 through 15 correspond to the author of the work, and the entire sequence 0 through 15 corresponds to the MARC 245 Title Statement for the work.

Text Classification

Text classification, on the other hand, is about characterizing the sentiment, subject, topic or theme of an entire text.

A book can have a particular subject, or be about a particular theme, or be an instance of a specific genre classification, without any of the words used to describe that subject heading or genre classification appearing anywhere in the book at all.

For example, we want our machine learning model to be able to identify this book as concerning **New York (N.Y.)—Social life and customs** whether or not those exact words appear in the book in that form.

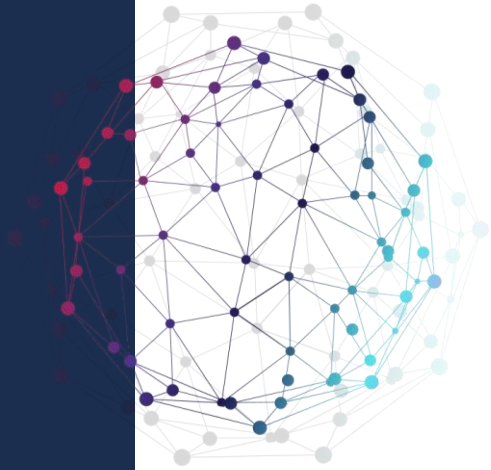
Successfully identifying the LCSH subject headings for an e-book text is an instance of particularly challenging problem in natural language processing sometimes described as Extreme Multi-label Text Classification (XML or XMLC).

Most ML text classification workflows operate with a few dozen or a few hundred classes. LCSH contains hundreds of thousands of classes.

The 23,000 e-books tested for this experiment alone contained over 26,000 subject headings.

Selection

Machine learning landscape



Core approaches

- **Natural Language Processing tools**: That is, widely used and supported NLP packages and tools such as Spacy, FlairNLP, NLTK, or Spark NLP designed for text classification, entity extraction, parts of speech tagging, and so on.
- **Existing bibliographic metadata extraction tools**: Particularly in the area of scientific documents and journal articles, there are a number of existing tools—such as Grobid, Cermine, or ParsCit—for extracting bibliographic metadata from PDFs.

Core approaches

- **Transformer based approaches**: Transformers are a type of deep learning model introduced by researchers at Google in a seminal 2017 paper: https://arxiv.org/pdf/1706.03762.pdf . Transformer based approaches currently lead the performance league tables for many natural language processing tasks such as data extraction, summarization, translation, etc.
- **Other library metadata tools**: Existing tools being used for subject classification within the library community, and especially at the level of national libraries or similar institutions.
- **Hybrid or multimodal approaches**: Which use information other than just the plaintext of the document, such as layout information or document structure to potentially improve the quality of results.

Deep Learning NLP models

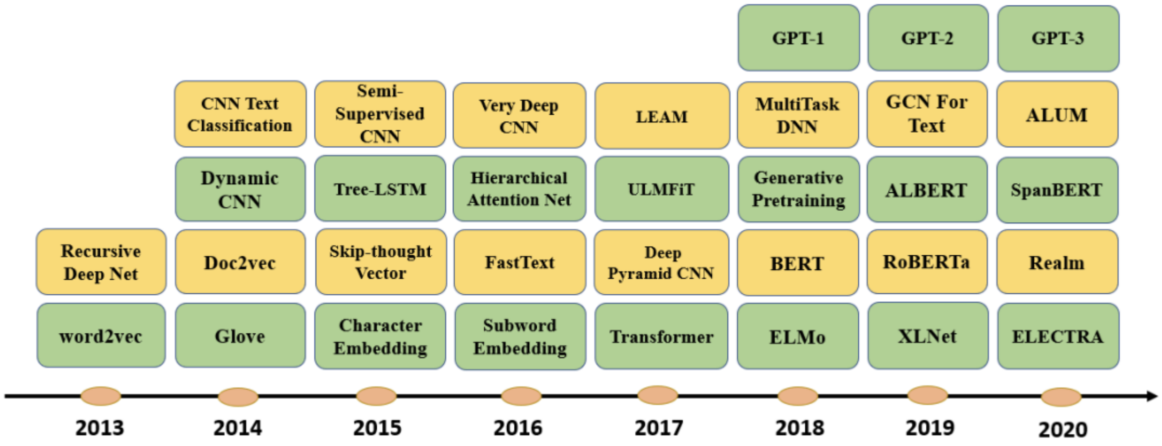


Fig. 22. Some of the most prominent deep learning models for text embedding and classification published from 2013 to 2020.

Figure above from [2004.03705] Deep Learning Based Text Classification: A Comprehensive Review

Deep Learning NLP models

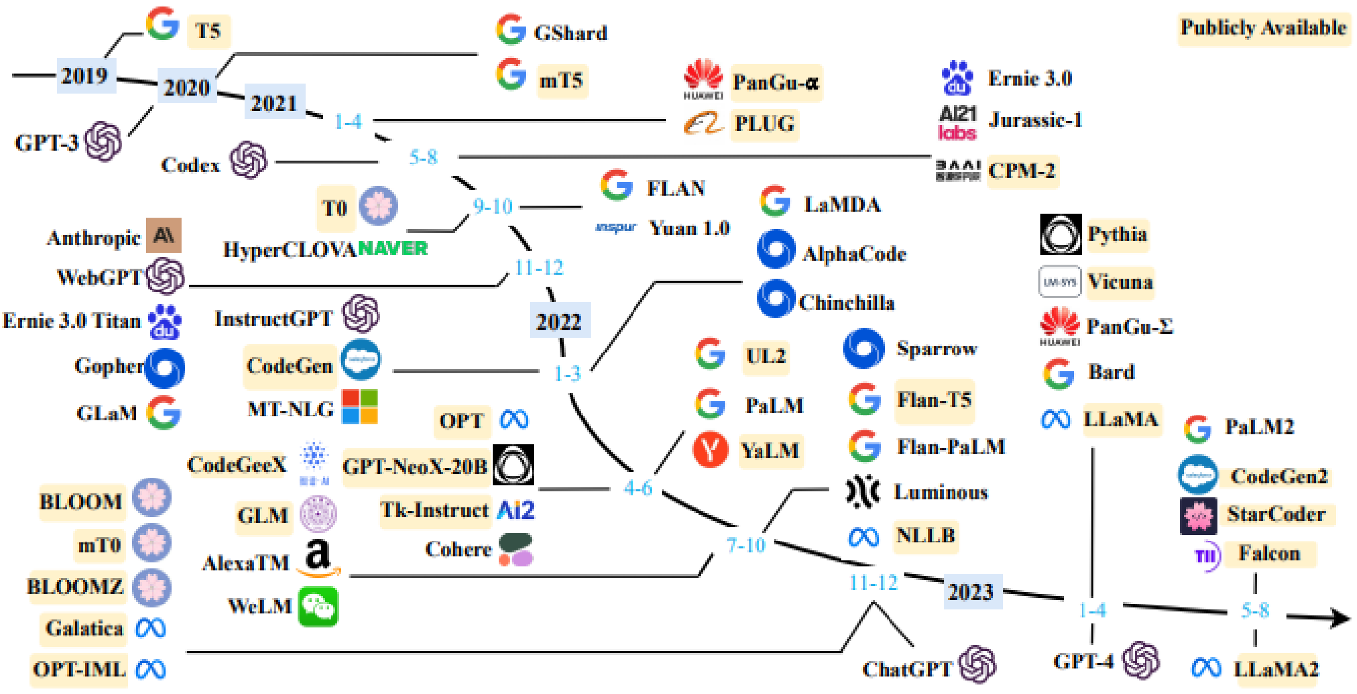
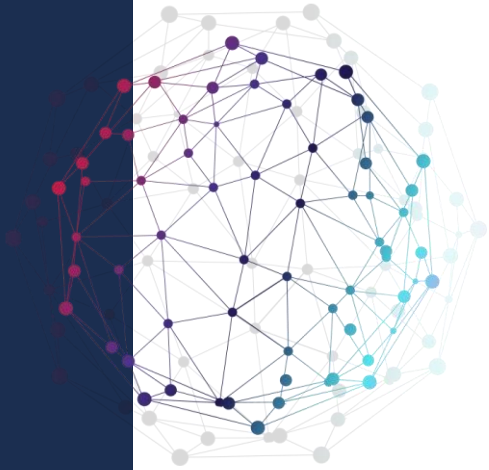


Figure above from [2303.18223] A Survey of Large Language Models

Selection

What similar activities are being employed by other organizations?



Text classification

A number of similar institutions are actively engaged in using machine learning models for text classification. Including:

- **National Library of Finland**: The original developers of Annif use it as part of their https://ai.finto.fi/ service. See: Annif and Finto AI: Developing and Implementing Automated Subject Indexing | JLIS.it
- **Deutsche National Bibliotek**: Annif is used to drive the German National Library's "indexing machine" for automatic content indexing, launched in April of 2022. See: DNB - AI-Project This indexing system is used to assign DDC subject groups and assign other text classification metadata.
- **National Library of Sweden**: For their subject classification service Datastatus | Swepub
- **ZBW (Leibniz Information Centre)**: Automation of Subject Indexing Using Methods from Artificial Intelligence | ZBW

Text classification

The institutions on the previous slide are largely consolidated around Annif, however, there are other approaches to text classification from similar organizations.

For example, the British Library has used crowd-sourced data to train a model to classify documents by genre.

See: https://huggingface.co/BritishLibraryLabs/bl-books-genre

Note, the BL Labs project was tested with a simple two label classification scheme (Fiction vs Nonfiction) whereas there are over 26,000 subjects in use on the 23,000 ebook training corpus for this project so the results are unlikely to generalize directly to the subject classification use case.

Token classification

There are a relatively large number of tools that parse and extract bibliographic metadata from PDFs or other formats. Most of these tools are designed—either as a secondary function, or as their primary purpose—to parse bibliographic references and citations from within the text.

- GROBID
- CERMINE
- ScienceParse
- ParseCit
- PdfAct

Most of these tools are primarily targeted at article length texts and especially scientific articles and pre-publication.

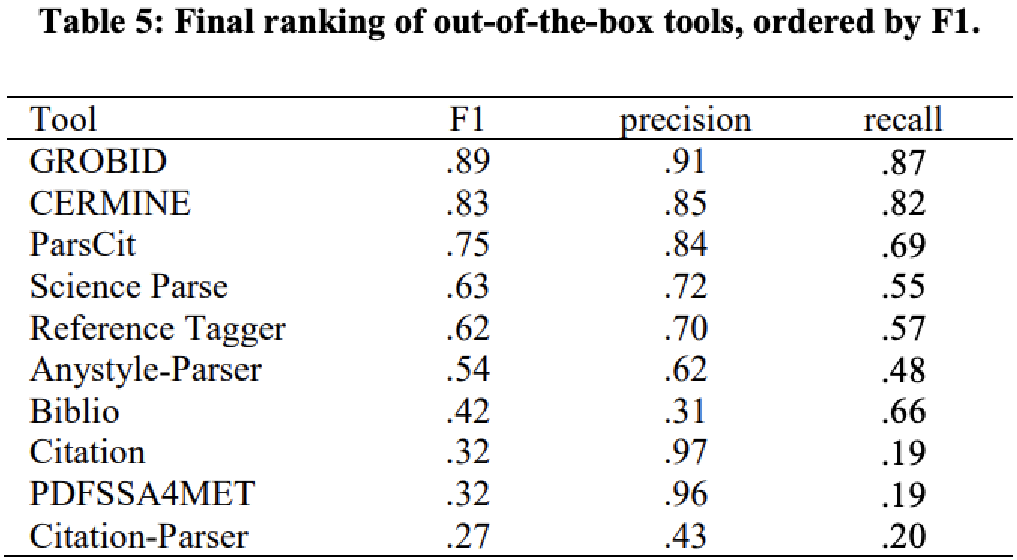


Table 5: Final ranking of out-of-the-box tools, ordered by F1.

Tool	F1	precision	recall
GROBID	.89	.91	.87
CERMINE	.83	.85	.82
ParsCit	.75	.84	.69
Science Parse	.63	.72	.55
Reference Tagger	.62	.70	.57
Anystyle-Parser	.54	.62	.48
Biblio	.42	.31	.66
Citation	.32	.97	.19
PDFSSA4MET	.32	.96	.19
Citation-Parser	.27	.43	.20

Figure from [1802.01168] Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers

Selection

What can we expect?

State of the Art: Token Classification

The best performing models tend to average 90-95% on the standard datasets used to evaluated token classification or named entity recognition such as the CONLL and Ontonotes data.

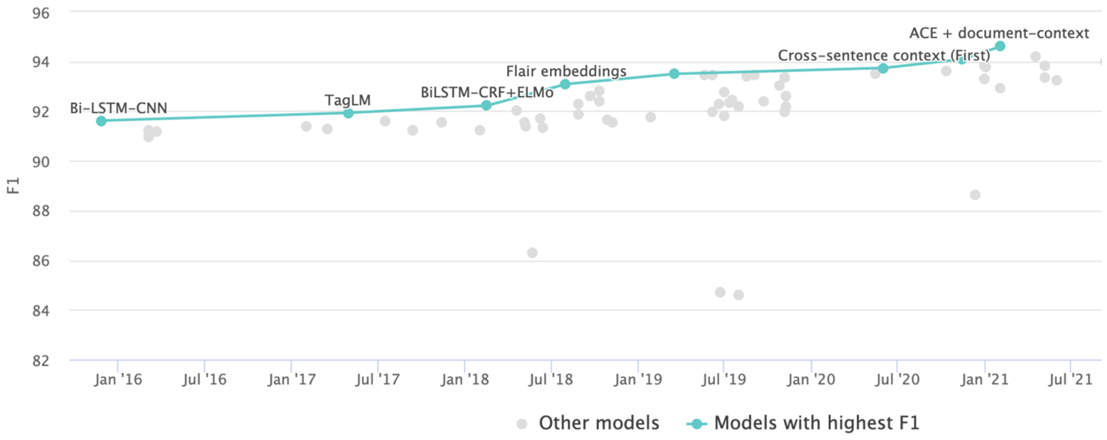


Figure from https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

Can we expect state of the art?

- ***Probably not***
- The CONLL and Ontonotes datasets and other similar datasets used as standard benchmarks have very short text extracts with very many instances of each of the classes of token / entity being trained and evaluated against
- There is over 20 years of concerted effort by NLP experts to achieve high scores on these datasets
- The classes in most token classification datasets are quite distinct:
    - Person
    - Place
    - Organization
- Whereas the MARC fields often overlap or are similar
- Following page shows evaluations from the bioinformatics / biomedical NLP domain which is probably a more realistic comparison

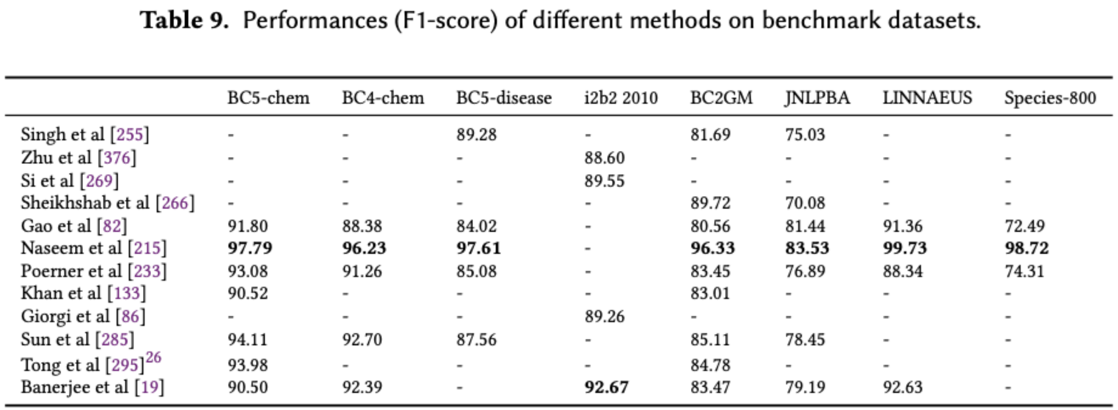


Table 9. Performances (F1-score) of different methods on benchmark datasets.

	BC5-chem	BC4-chem	BC5-disease	i2b2 2010	BC2GM	JNLPBA	LINNAEUS	Species-800
Singh et al [255]	-	-	89.28	-	81.69	75.03	-	-
Zhu et al [376]	-	-	-	88.60	-	-	-	-
Si et al [269]	-	-	-	89.55	-	-	-	-
Sheikhshab et al [266]	-	-	-	-	89.72	70.08	-	-
Gao et al [82]	91.80	88.38	84.02	-	80.56	81.44	91.36	72.49
Naseem et al [215]	**97.79**	**96.23**	**97.61**	-	**96.33**	**83.53**	**99.73**	**98.72**
Poerner et al [233]	93.08	91.26	85.08	-	83.45	76.89	88.34	74.31
Khan et al [133]	90.52	-	-	-	83.01	-	-	-
Giorgi et al [86]	-	-	-	89.26	-	-	-	-
Sun et al [285]	94.11	92.70	87.56	-	85.11	78.45	-	-
Tong et al [295][26]	93.98	-	-	-	84.78	-	-	-
Banerjee et al [19]	90.50	92.39	-	**92.67**	83.47	79.19	92.63	-

Table from [2110.05006] Pre-trained Language Models in Biomedical Domain: A Systematic Survey

- Figures from the domain specific and more challenging token classification tasks tend to range between 40% and 95%
- The best performing models tend to average around 80% across multiple datasets

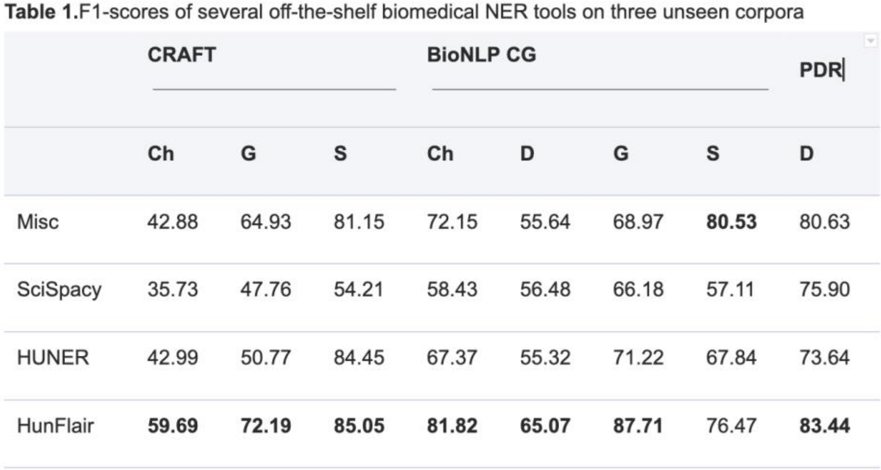


Table 1.F1-scores of several off-the-shelf biomedical NER tools on three unseen corpora

	CRAFT			BioNLP CG				PDR
	Ch	G	S	Ch	D	G	S	D
Misc	42.88	64.93	81.15	72.15	55.64	68.97	**80.53**	80.63
SciSpacy	35.73	47.76	54.21	58.43	56.48	66.18	57.11	75.90
HUNER	42.99	50.77	84.45	67.37	55.32	71.22	67.84	73.64
HunFlair	**59.69**	**72.19**	**85.05**	**81.82**	**65.07**	**87.71**	76.47	**83.44**

Table from https://academic.oup.com/bioinformatics/article/37/17/2792/6122692

80%+

Is probably a reasonable expectation for performance on most non-subject MARC fields

State of the Art: Text Classification

The best performing models tend to produce similar scores on some of the common text classification datasets. The graph below shows error so lower is better.

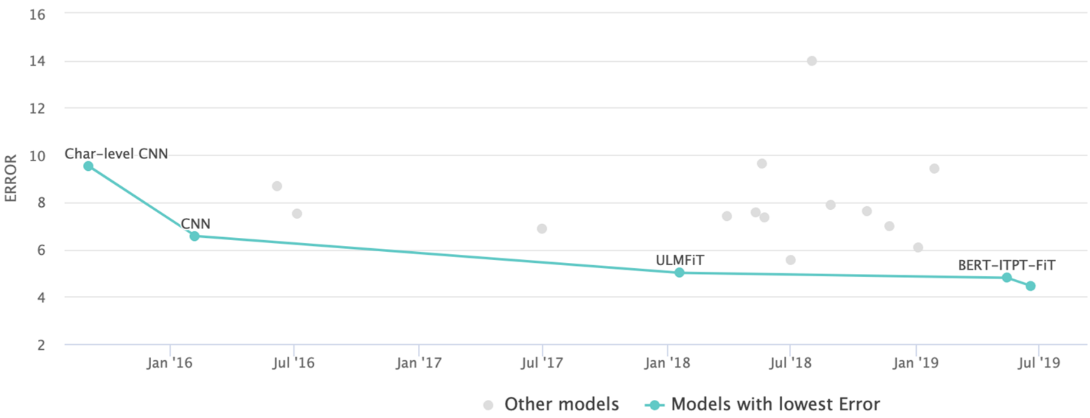


Figure from https://paperswithcode.com/sota/text-classification-on-ag-news

Can we expect state of the art?

- ***Certainly not***
- Most standard text classification datasets tend to have a few dozen or at most a few hundred classes
- Most standard text classification datasets tend to have relatively short texts
- The following page shows the distribution of labels in Extreme Multi-label classification datasets

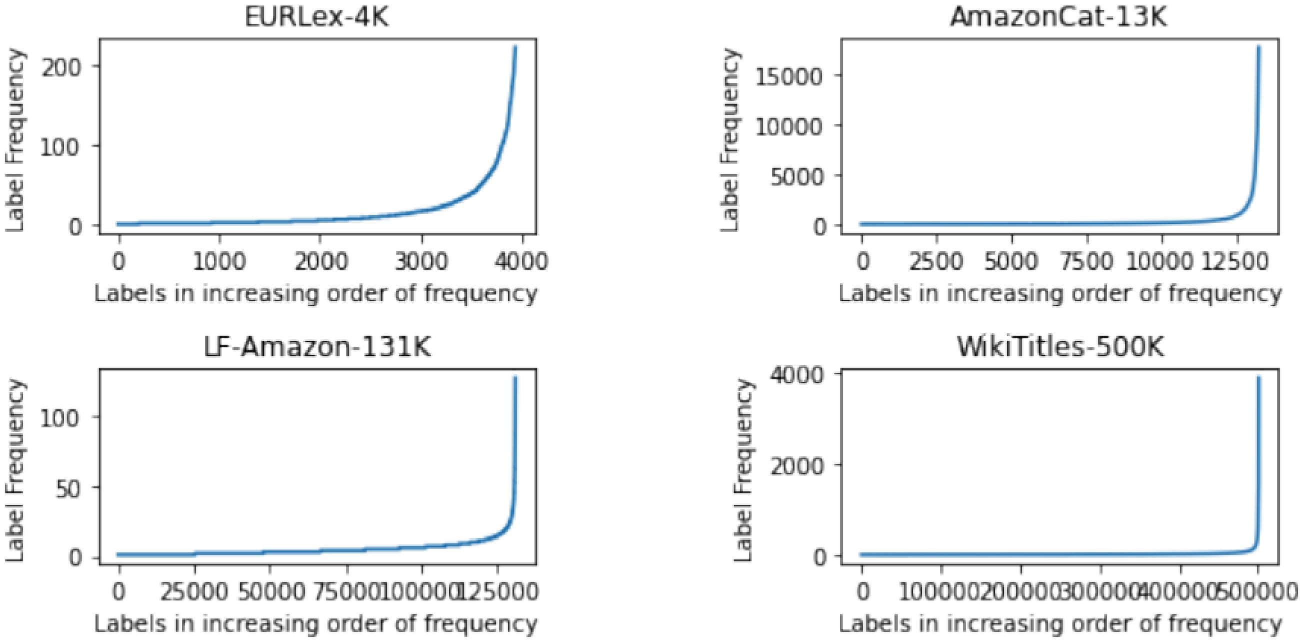
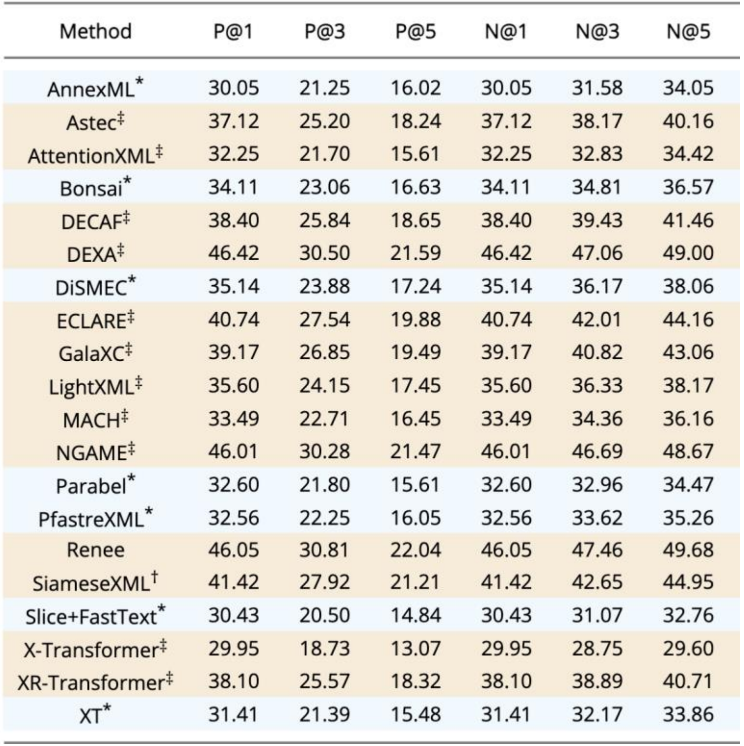


Figure from Dasgupta, A., Katyan, S., Das, S., & Kumar, P. (2023). Review of Extreme Multilabel Classification. *arXiv preprint arXiv:2302.05971*.

Expectations: Text Classification



Method	P@1	P@3	P@5	N@1	N@3	N@5
AnnexML*	30.05	21.25	16.02	30.05	31.58	34.05
Astec‡	37.12	25.20	18.24	37.12	38.17	40.16
AttentionXML‡	32.25	21.70	15.61	32.25	32.83	34.42
Bonsai*	34.11	23.06	16.63	34.11	34.81	36.57
DECAF‡	38.40	25.84	18.65	38.40	39.43	41.46
DEXA‡	46.42	30.50	21.59	46.42	47.06	49.00
DiSMEC*	35.14	23.88	17.24	35.14	36.17	38.06
ECLARE‡	40.74	27.54	19.88	40.74	42.01	44.16
GalaXC‡	39.17	26.85	19.49	39.17	40.82	43.06
LightXML‡	35.60	24.15	17.45	35.60	36.33	38.17
MACH‡	33.49	22.71	16.45	33.49	34.36	36.16
NGAME‡	46.01	30.28	21.47	46.01	46.69	48.67
Parabel*	32.60	21.80	15.61	32.60	32.96	34.47
PfastreXML*	32.56	22.25	16.05	32.56	33.62	35.26
Renee	46.05	30.81	22.04	46.05	47.46	49.68
SiameseXML†	41.42	27.92	21.21	41.42	42.65	44.95
Slice+FastText*	30.43	20.50	14.84	30.43	31.07	32.76
X-Transformer‡	29.95	18.73	13.07	29.95	28.75	29.60
XR-Transformer‡	38.10	25.57	18.32	38.10	38.89	40.71
XT*	31.41	21.39	15.48	31.41	32.17	33.86

- Performance on most metrics, for Extreme Multi-label Text Classification tend to hover around 20-30% for @5 metrics (where the top 5 ranked classes are being evaluated)
- Most of these XMLC algorithms or models were trained and tested on much larger datasets.
- Research into intersubjective agreement between expert human catalogers has shown that catalogers often only agree exactly on subject classification around 30-50% of the time. https://researchcommons.waikato.ac.nz/handle/10289/3513

See: http://manikvarma.org/downloads/XC/XMLRepository.html#benchmarks for a full set of benchmarks

30%

Is probably a reasonable upper expectation for performance on LCSH classification

What did we select and why?

GROBID

GROBID (GeneRation Of BIbliographic Data) is a machine learning library for extracting, parsing and re-structuring raw documents such as PDF into structured XML/TEI encoded documents with a particular focus on technical and scientific publications.

GROBID is widely used in a number of institutions, including: ResearchGate, HAL Research Archive, the European Patent Office, INIST-CNRS, Mendeley, CERN (Invenio), and the Internet Archive.

GROBID

GROBID was chosen because:

- GROBID can convert PDF to text/XML files, so the outputs from GROBID can be reused to provide data for the other models, including Model 2, 3, 4, and 5.
- GROBID can extract bibliographic information
- GROBID scores highly on reported metrics
- GROBID is well documented and actively maintained
- GROBID can extract structural information and coordinate information as part of the PDF processing, which could be reused for Model 5.

We were aware of some risks at selection time:

- GROBID is not ideally suited to longer texts
- GROBID is typically trained with a small corpus of annotated TEI-XML documents and there were some potential risks about generating suitable training documents from the existing MARCXML records.

Annif

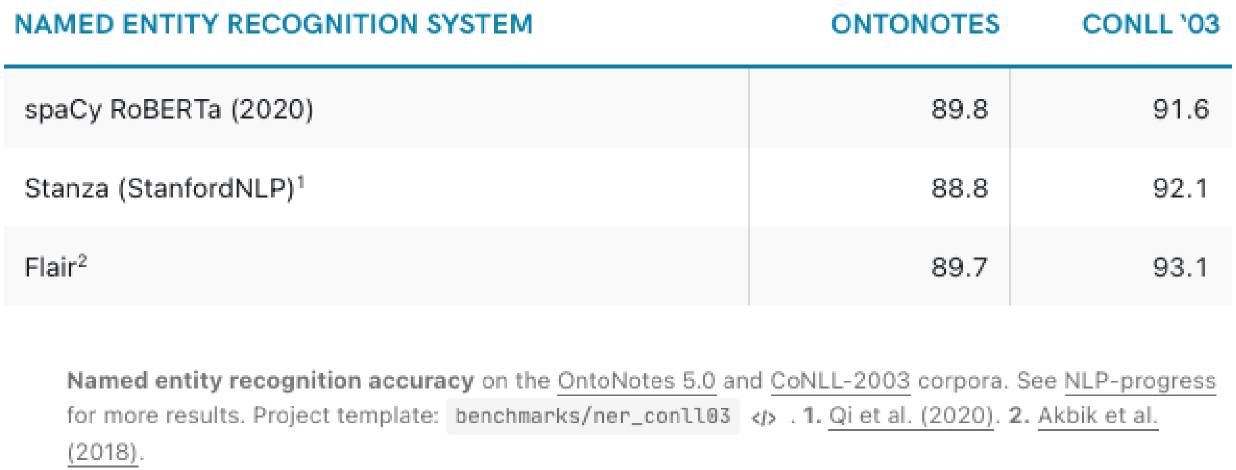
- **Good scores on published metrics**
- **Diversity of technology**: Annif wraps multiple backends allowing us to test multiple different approaches from the research literature on extreme multi-label classification.
- **Developer friendliness and code quality**: Annif has a simple command line interface and configuration management system so can be deployed and tested easily. The configuration system is flexible while remaining straightforward. The code is in Python and it is possible to configure a local development install easily, by inheriting from an existing base class in the code.
- **Quality and comprehensiveness of documentation**: Annif has excellent documentation and a Wiki.
- **Project activity and responsiveness to issues**: Annif is regularly updated.
- **Ease of generating training data**: The tab delimited format used by Annif is very easy to create and to work with documents at scale.

Spacy

- **Diversity of technology**: Spacy provides multiple language models, and has integrations with its own trained models and with Transformer based models available via HuggingFace. Spacy can also integrate easily with external tools and pipelines, and can be extended. While it came too late for the testing on this project, Spacy has recently launched integrations with generative AI models such as ChatGPT and Llama.
- **Developer friendliness and code quality**: Spacy provides a project based command line interface and configuration management system so can be deployed and tested easily. It can also be easily integrated via the Python API.
- **Quality and comprehensiveness of documentation**: High quality comprehensive documentation and example projects.
- **Project activity and responsiveness to issues**: Very regularly updated with new features.
- **Ease of generating training data**: Less easy than Annif, but works with JSONLines files and their own binary DocBin format (which is easy to create and compact).

Spacy

- **Performance on published metrics**



NAMED ENTITY RECOGNITION SYSTEM	ONTONOTES	CONLL '03
spaCy RoBERTa (2020)	89.8	91.6
Stanza (StanfordNLP)[1]	88.8	92.1
Flair[2]	89.7	93.1

Named entity recognition accuracy on the OntoNotes 5.0 and CoNLL-2003 corpora. See NLP-progress for more results. Project template: `benchmarks/ner_conll03` </> . **1.** Qi et al. (2020). **2.** Akbik et al. (2018).

Hugging Face: Transformers

BERT, launched in 2017 via the influential paper [1706.03762] Attention Is All You Need - arXiv introduced the Transformer architecture and led to the rise of Large Language Models (such as OpenAI’s ChatGPT and Meta’s Llama).

The leaderboards in almost every area of natural language processing are dominated by Transformer based models and/or Large Language Models (which are also typically examples of the same broad architecture) or by models which leverage some aspect of Transformers as part of their workflow..

Hugging Face: Transformers

- **Performance on published metrics**: Transformer models or related large language models, are typically the state of the art on many NLP tasks, including information extraction / entity extraction and text classification, which were the two core tasks being tested as part of this experiment.
- **Diversity of technology**: With a number of different pretrained models available via HuggingFace we could test a number of models using the same core code base and take advantage of existing training via fine-tuning.
- **Quality and comprehensiveness of documentation**: High quality comprehensive documentation and example project for most of our core use cases
- **Ease of generating training data**: We were able to reuse the training data generated for training Spacy, via straightforward format conversion.

Spacy with Positional Data

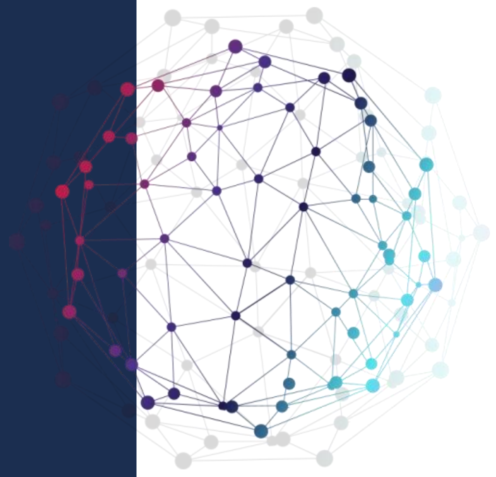
Our **fifth** experiment/model was intended to examine whether we could leverage additional information about document layout, text size, text position on the page, etc alongside the content of the text to improve the quality of our results.

Intuitively, bibliography metadata sometimes carries distinctive properties such as larger text for titles, or a particular style of text block for the CIP (Cataloging-In-Publication) block found in many ebooks.

Our intention was to take the best performing, or pragmatically the easiest to work with, model from experiment 3 (Spacy) and 4 (BERT/Transformers) and then add an additional layer in the training process that leveraged bounding box information to try to improve the results.

Selection

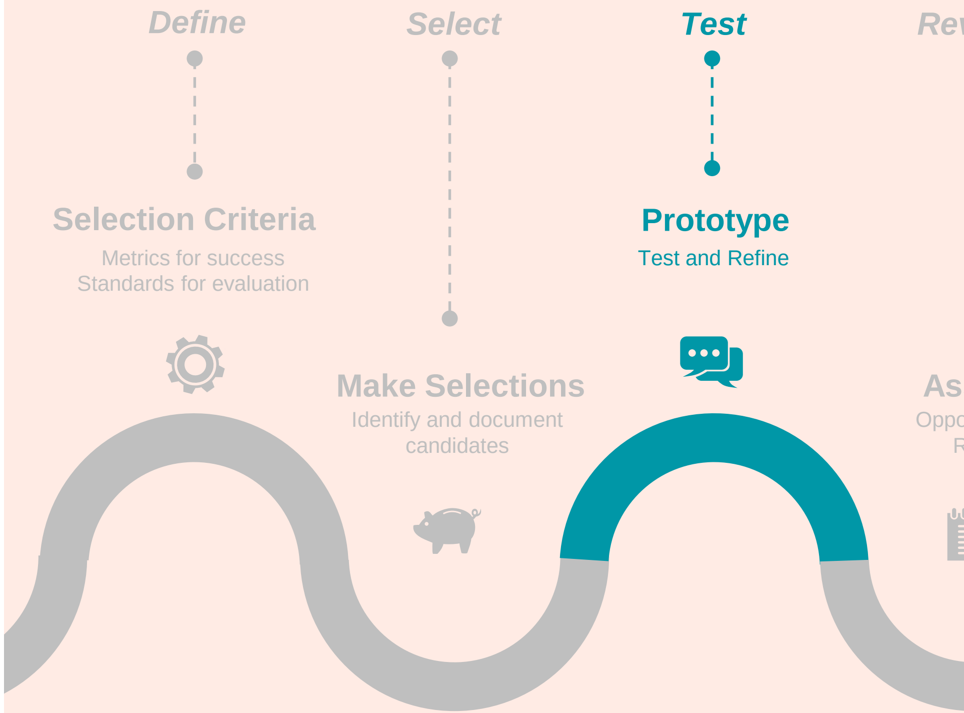
Why not ChatGPT?



- ChatGPT launched after the initial selection of models was made for this experiment
- GPT-4 launched in March of 2023 at which point we had already begun the assisted cataloging prototypes
- No APIs were available for passing data to ChatGPT until spring of 2023 (this year)
- No method existed for testing ChatGPT (GPT-3.5 or GPT-4) until after we had completed the prototyping for the core 5 machine learning experiments
- Models using the same underlying Transformer architecture were already in our set of prototypes

However, given the attention being paid to OpenAI's generative AI models, we did do limited testing of GPT-3.5 and a non-commercial alternative (Meta's LLama-2) as part of a “sixth” model.

Prototyping



Define

Selection Criteria

Metrics for success
Standards for evaluation

Select

Make Selections

Identify and document
candidates

Test

Prototype

Test and Refine

Rev

As

Oppo
R

Prototypes

Generative AI

As a **sixth** additional prototype, we tested two generative AI models:

- ChatGPT (GPT-3.5)
- Llama-2 (13b parameter version)

Each was few the first 3,000 words of the e-book text with a prompt to return bibliographic metadata for the core fields.

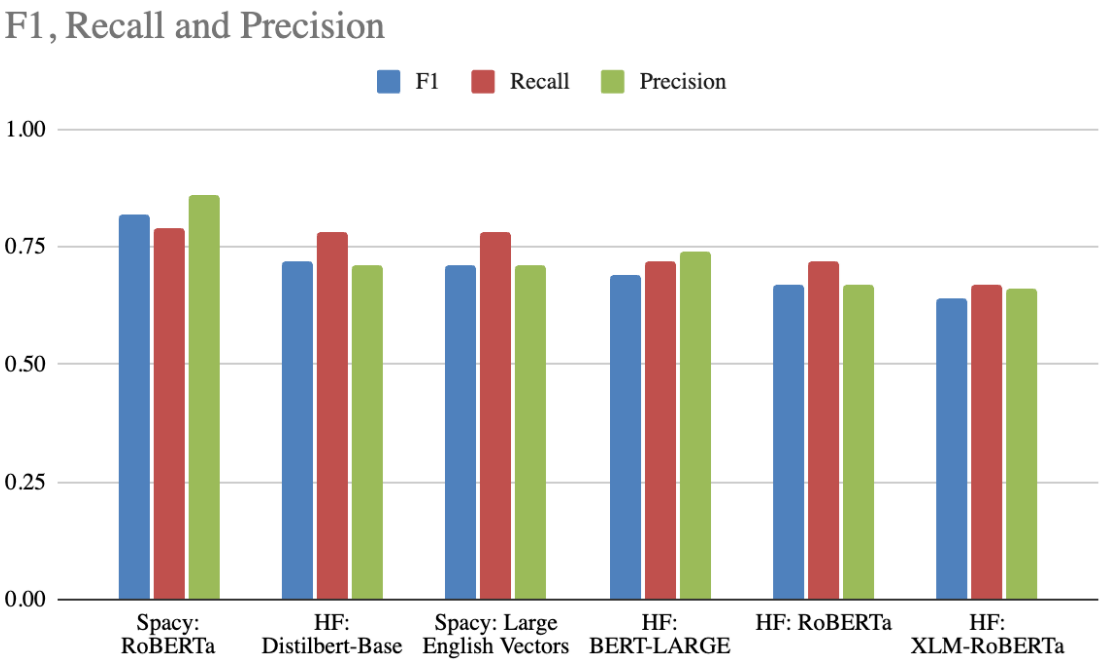
Metrics were calculated based on:

- Title
- Author

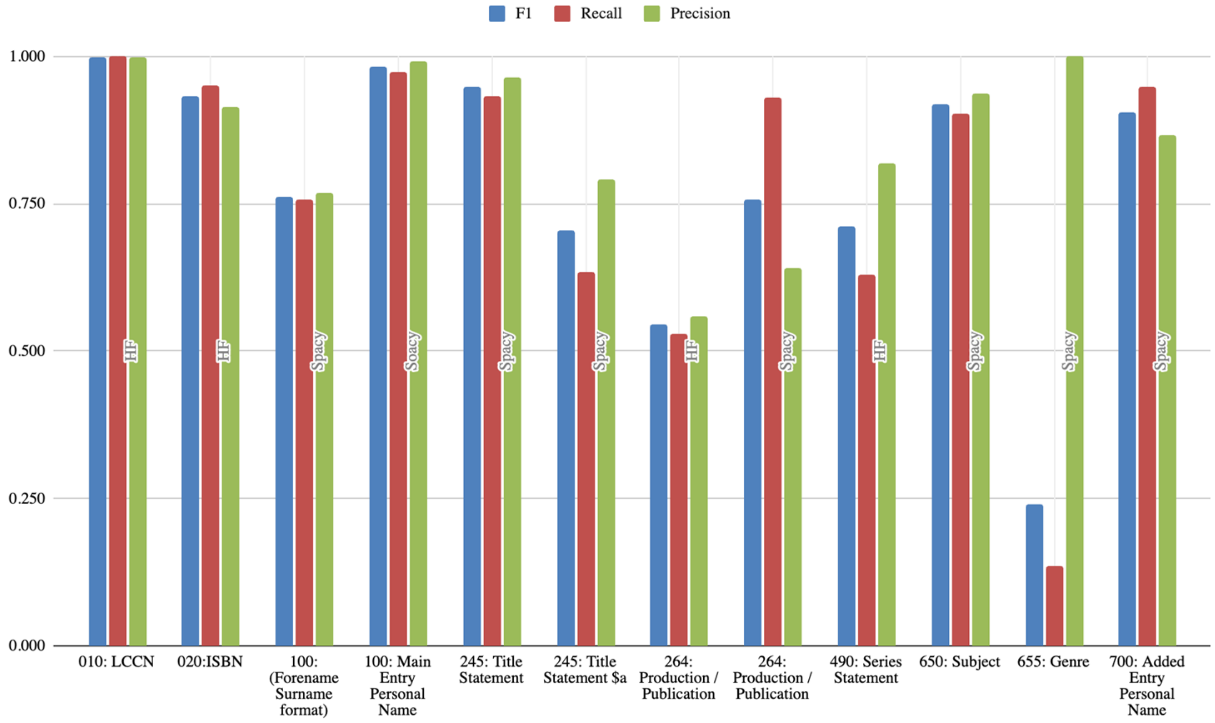
… and compared to similar metrics for the Hugging Face and Spacy frameworks tested using the same method, i.e. extracting fields from the first 3,000 words of the e-book.

Prototypes

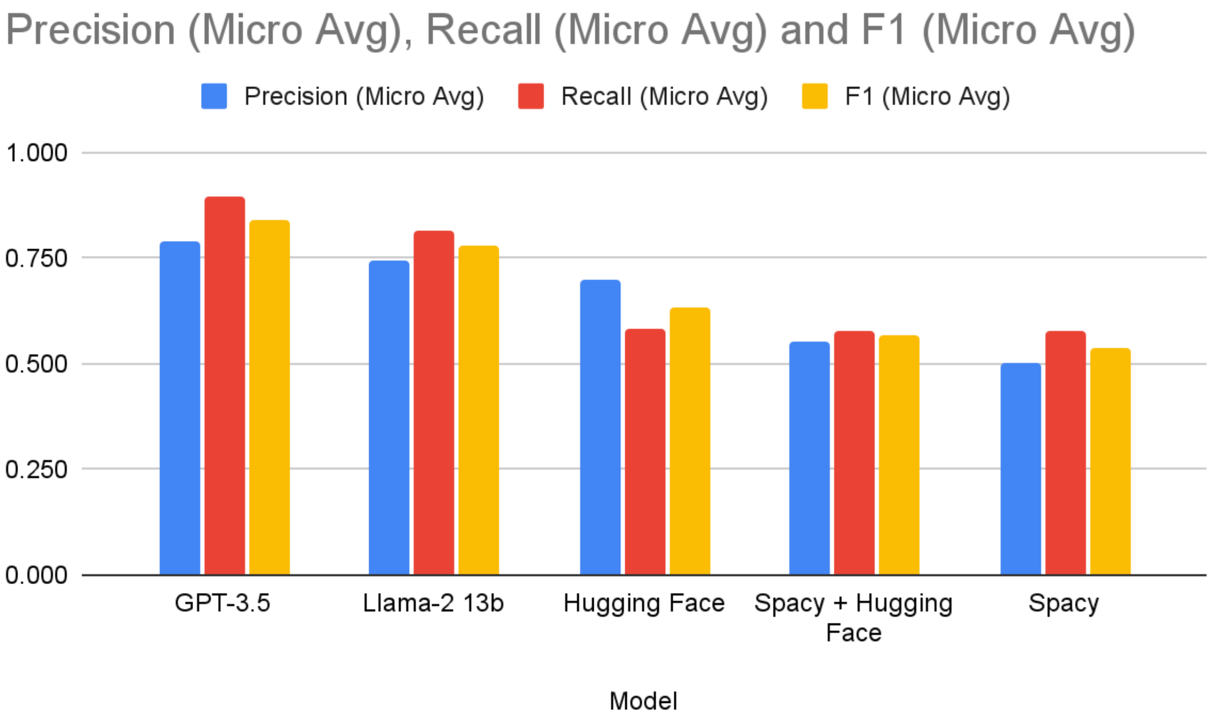
Findings: Performance



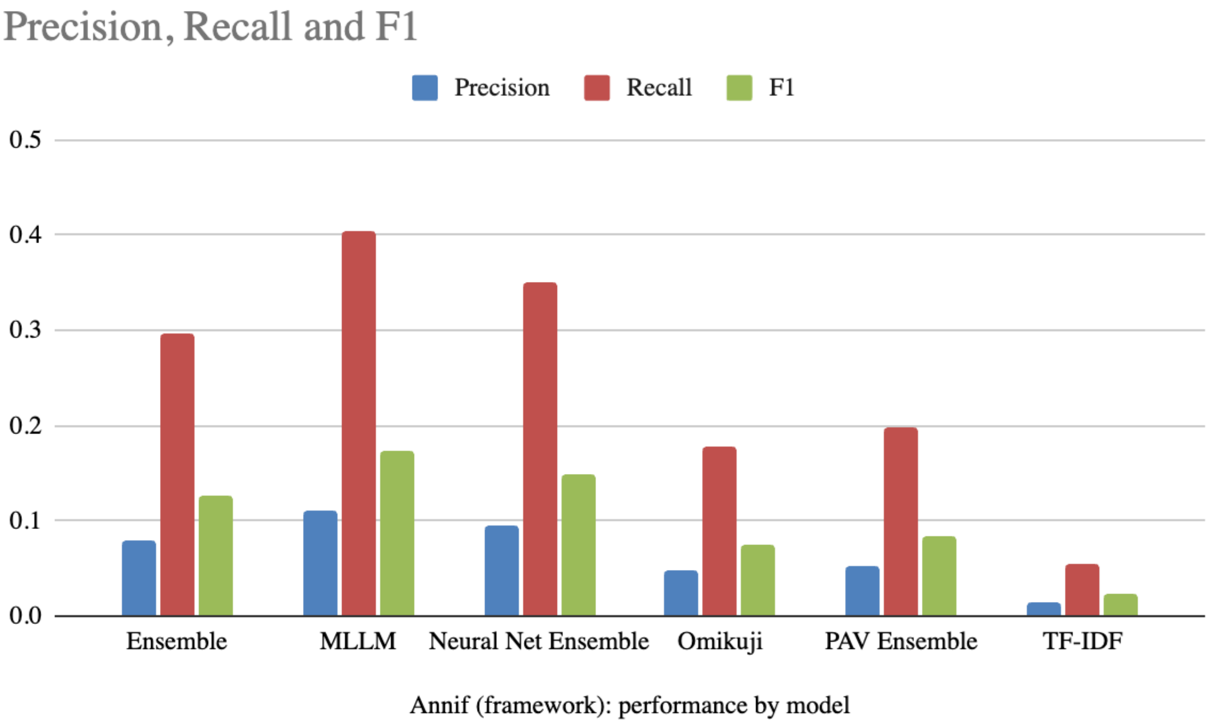
Token Classification Metrics, averages across all fields. By prototype/framework and language model.



Token Classification Metrics, performance of best performing framework: model combination, per field



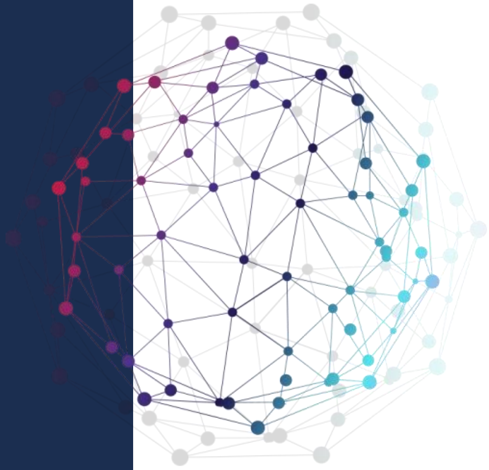
Token Classification Metrics, averages for Title and Author. By prototype/framework, best performing model chosen for each.



Subject Classification with Annif: Scores for different models, tested on the first 5,000 words of each e-book

Prototypes

Findings: Manual Review



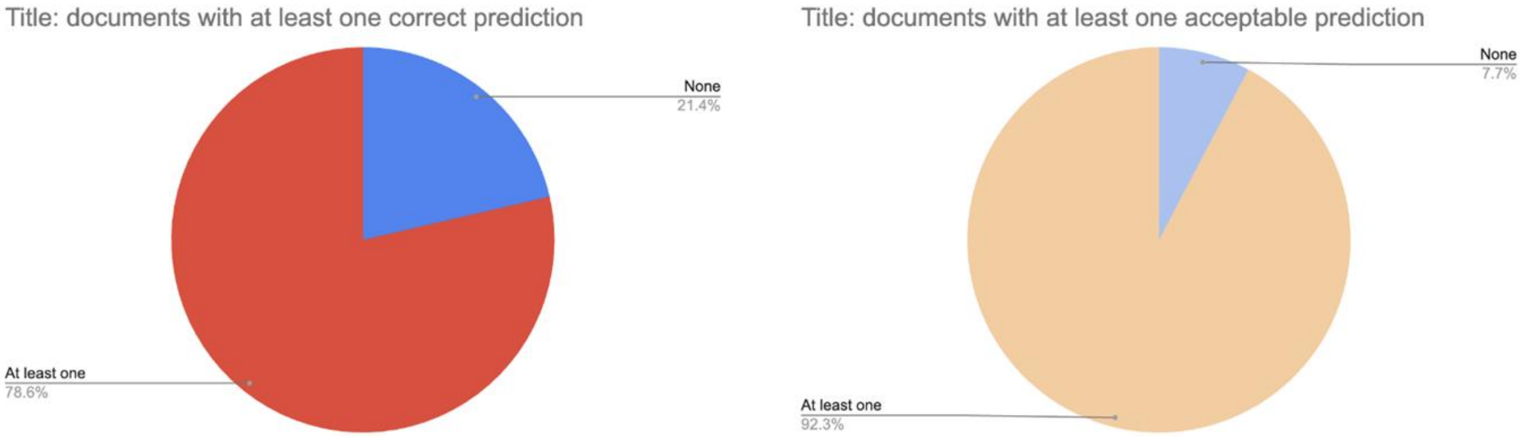
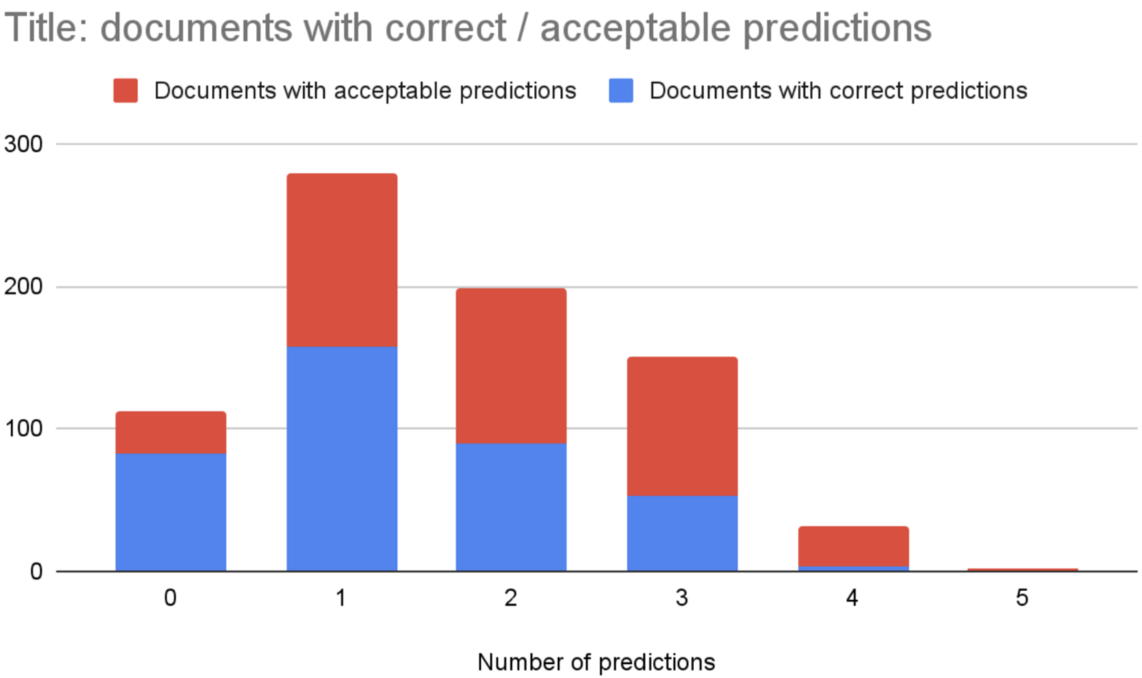


Figure: Acceptable versus Correct MARC 245


Title: documents with correct / acceptable predictions
🟥 Documents with acceptable predictions
🟦 Documents with correct predictions
300
200
100
0
0
1
2
3
4
5
Number of predictions

Manual Review

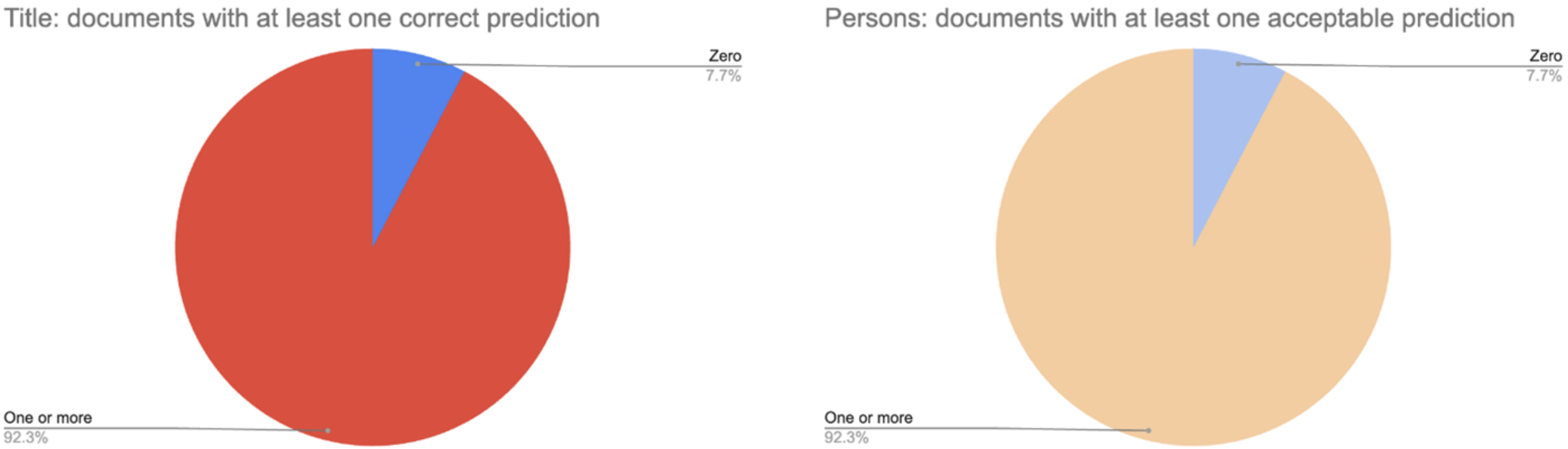
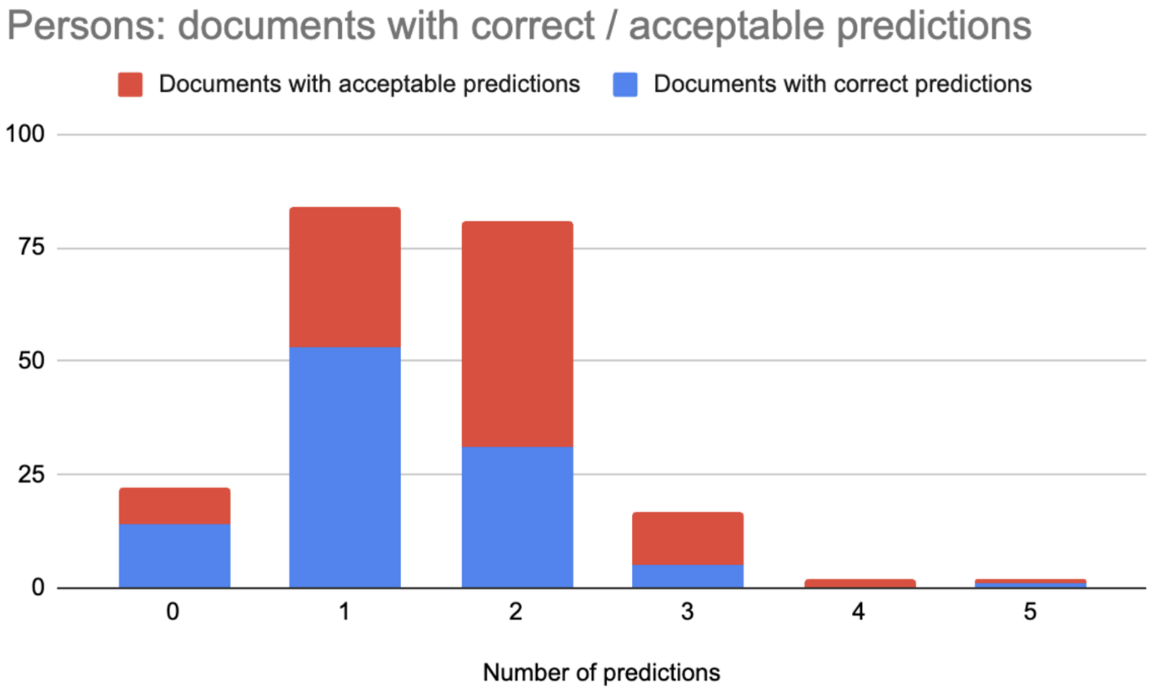


Figure: Acceptable versus Correct MARC 100 and 700 fields


Persons: documents with correct / acceptable predictions
🟥 Documents with acceptable predictions
🟦 Documents with correct predictions
100
75
50
25
0
0
1
2
3
4
5
Number of predictions

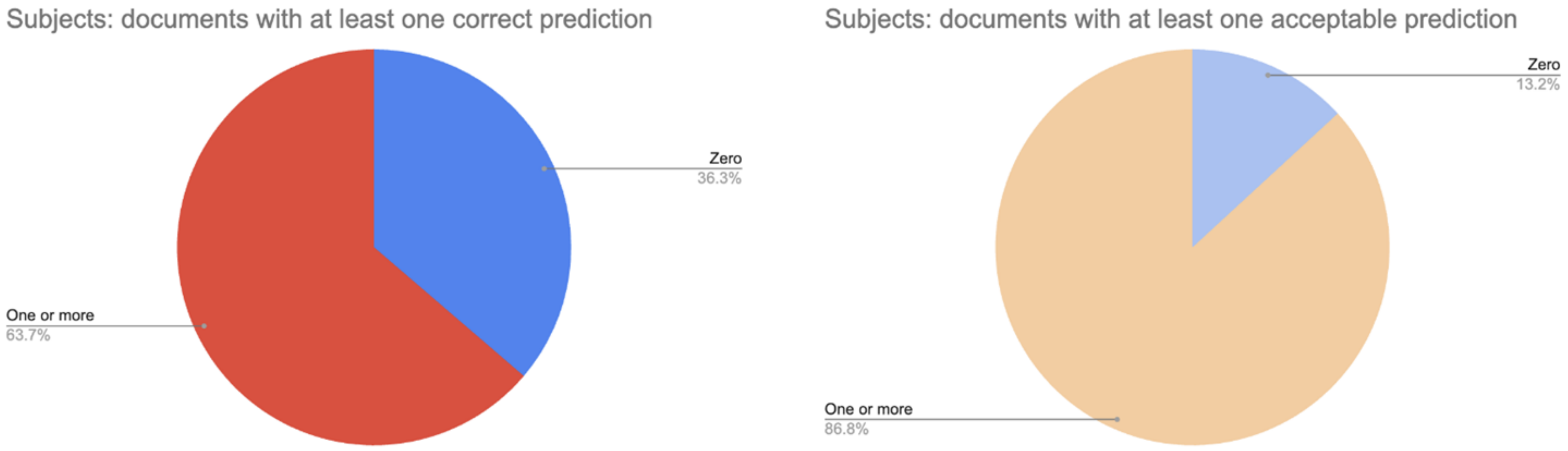
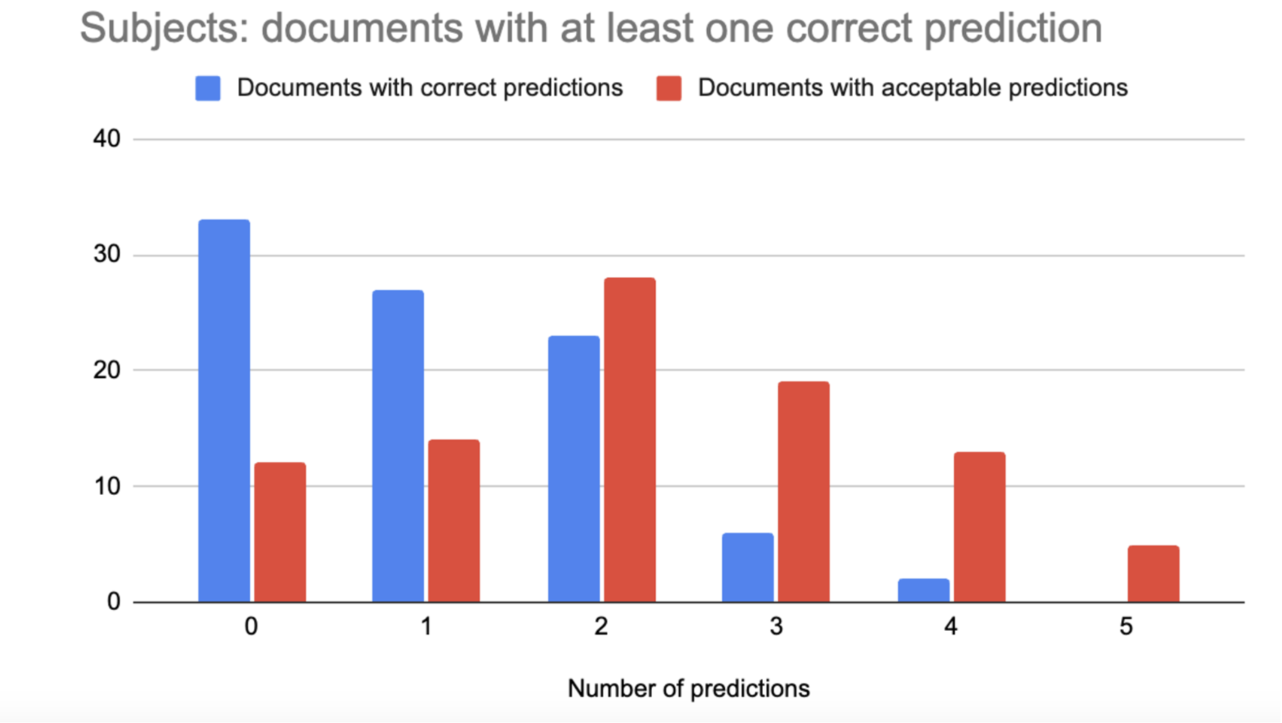


Figure: Acceptable versus Correct MARC 6xx fields

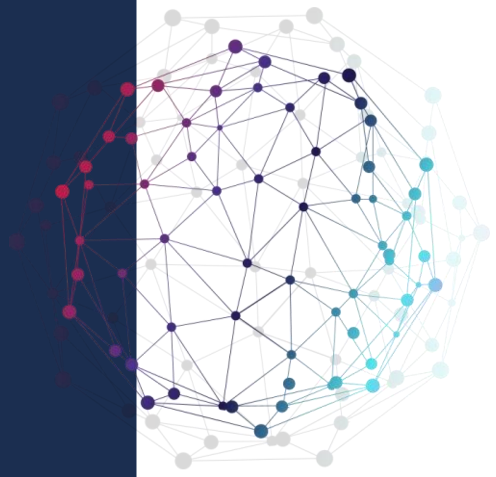

Subjects: documents with at least one correct prediction
🟦 Documents with correct predictions
🟥 Documents with acceptable predictions
40
30
20
10
0
0
1
2
3
4
5
Number of predictions

What we learned

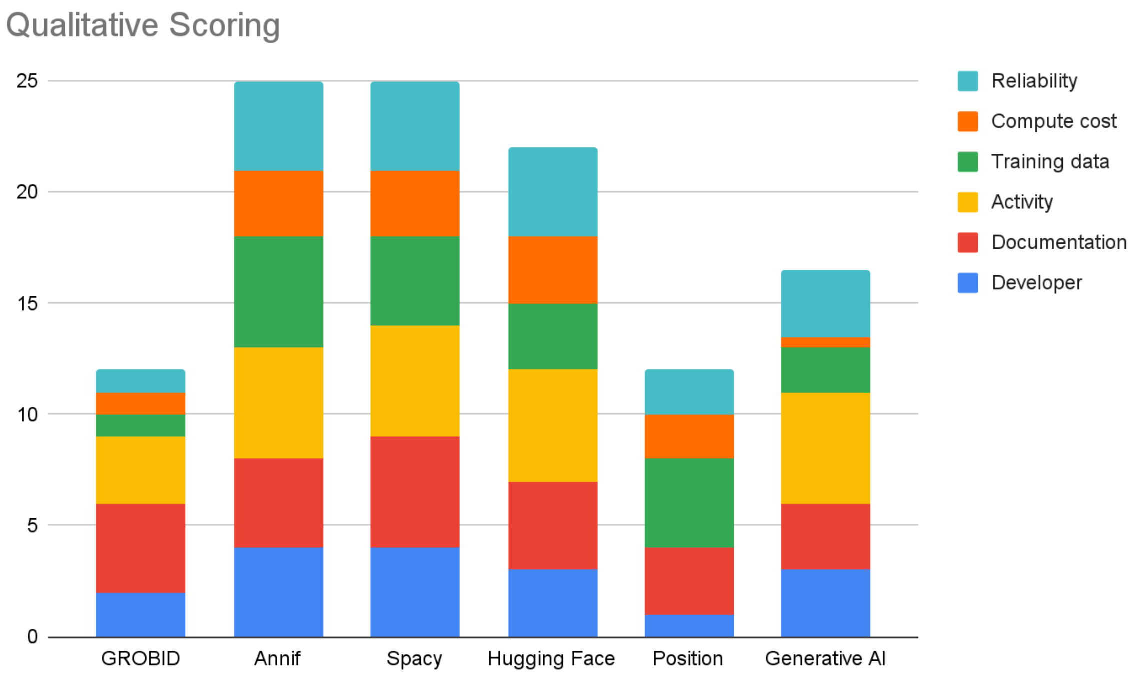
- Token classification approaches—Title, Person—generally generated one or more correct predictions for around 80% - 90% of documents
- For most documents the number of acceptable documents was higher as the incorrect predictions were often partial matches or close variants.
- Text classification approaches—Subject Headings—were generally less successful, and there was more variation between documents.
- At least one correct or acceptable prediction was generated for the majority of documents
- There was more variation between the number of acceptable and correct Subject Headings as a higher percentage of incorrect Subject Headings were judged acceptable

Prototypes

Findings: Practical



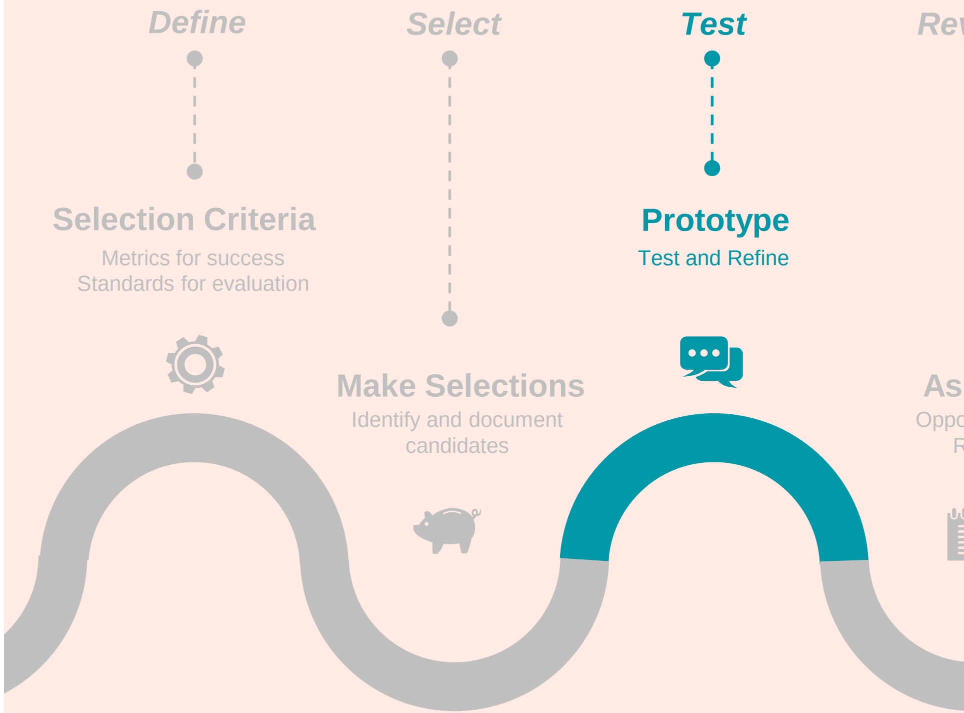
Ranking versus selection criteria



Practical usability

- Spacy, Annif and the Hugging Face libraries are all well documented, up to date, are developer friendly, and easy to use.
- GROBID is much harder to generate training data, and proved both unreliable and costly in practice.
- The positional model was an internal experiment, so naturally had less documentation, and no prior art to draw from.
- The generative AI approaches are generally quite well documented in terms of the APIs and prompting formatted, but:
    - Integration with other tooling is still relatively new
    - Training and fine-tuning is challenging
    - Evaluating them vis a vis the E-Book data required a lot of bespoke code
    - Compute requirements for self-hosted pipeline are very high

Prototyping: Assisted Cataloging



Define

Selection Criteria

Metrics for success
Standards for evaluation

Select

Make Selections

Identify and document
candidates

Test

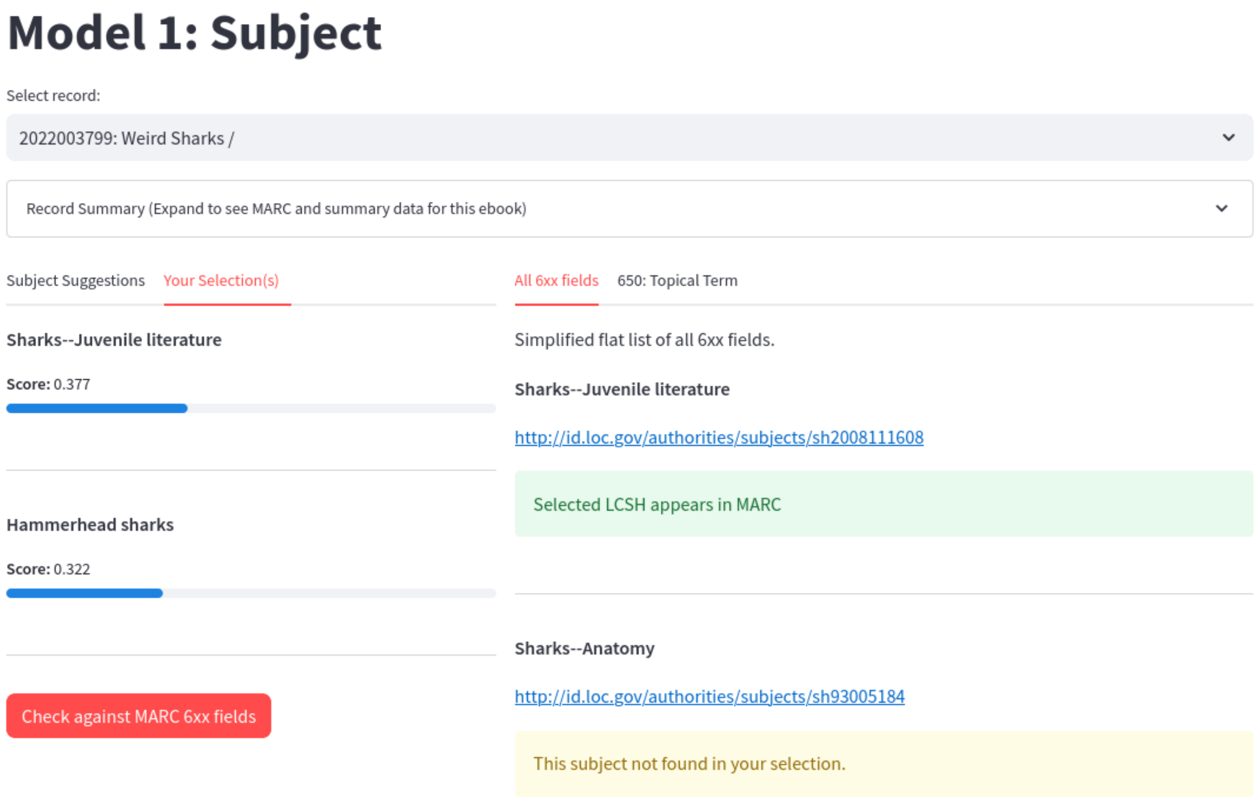
Prototype

Test and Refine

Rev

As

Oppo
R



Model 1: Subject

Select record:

2022003799: Weird Sharks /

Record Summary (Expand to see MARC and summary data for this ebook)

Subject Suggestions | Your Selection(s)

Sharks--Juvenile literature

Score: 0.377

Hammerhead sharks

Score: 0.322

Check against MARC 6xx fields

All 6xx fields | 650: Topical Term

Simplified flat list of all 6xx fields.

Sharks--Juvenile literature

http://id.loc.gov/authorities/subjects/sh2008111608

Selected LCSH appears in MARC

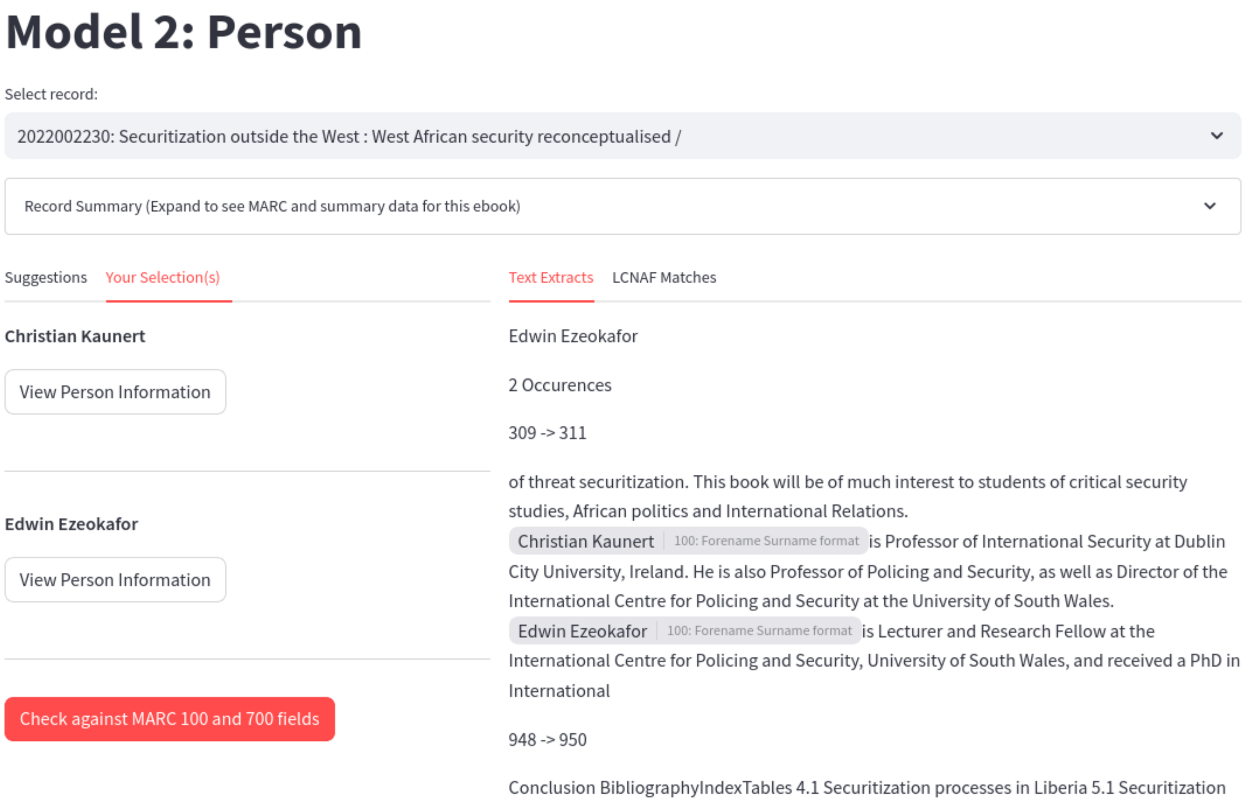
Sharks--Anatomy

http://id.loc.gov/authorities/subjects/sh93005184

This subject not found in your selection.

We were testing a number of hypotheses for assisting catalogers with subject classification:

- Whether keywords would be useful
- Whether abstractive or extractive summaries would be useful
- The value of retrieving LCSH data from id.loc.gov for use by catalogers



Model 2: Person

Select record:

2022002230: Securitization outside the West : West African security reconceptualised /

Record Summary (Expand to see MARC and summary data for this ebook)

Christian Kaunert

Edwin Ezeokafor

Edwin Ezeokafor

2 Occurences

309 -> 311

of threat securitization. This book will be of much interest to students of critical security studies, African politics and International Relations. Christian Kaunert 100: Forename Surname format is Professor of International Security at Dublin City University, Ireland. He is also Professor of Policing and Security, as well as Director of the International Centre for Policing and Security at the University of South Wales. Edwin Ezeokafor 100: Forename Surname format is Lecturer and Research Fellow at the International Centre for Policing and Security, University of South Wales, and received a PhD in International

948 -> 950

Conclusion BibliographyIndexTables 4.1 Securitization processes in Liberia 5.1 Securitization

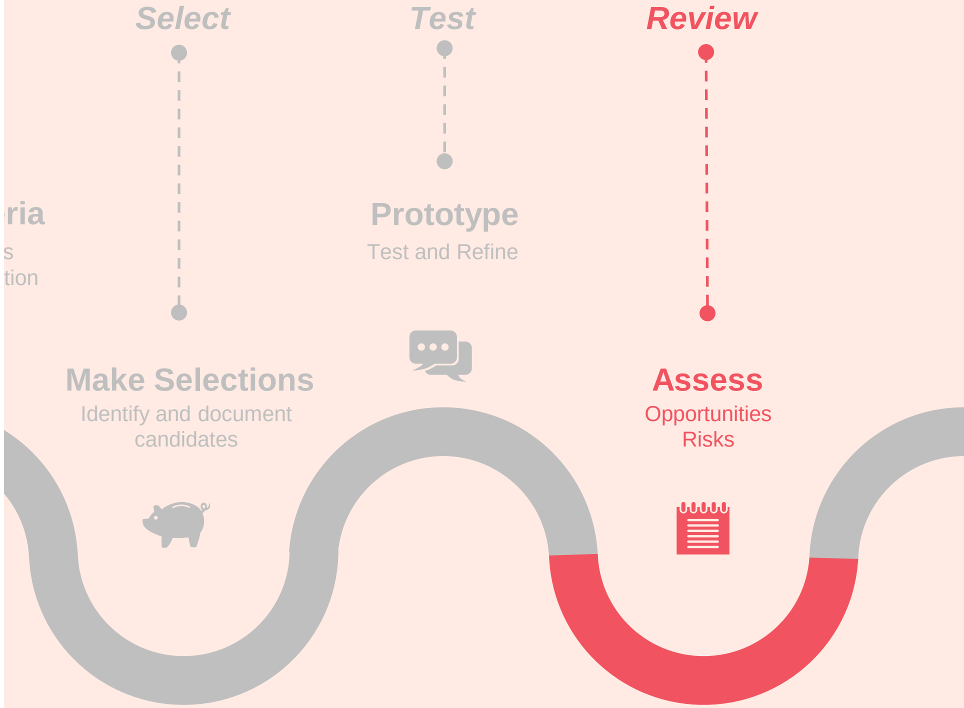
We were testing a number of hypotheses for assisting catalogers with cataloging people:

- Whether abstractive or extractive summaries would be useful
- The value of retrieving LCNAF data from id.loc.gov for use by catalogers
- The value of showing the cataloger the term in the context of the text of the e-book

What We Learned

- The MARC record as an assisting element scored nearly unanimous positive reviews.
- Summaries (abstractive, extractive) did not score highly, and were sometimes assessed as too long.
- Keywords were potentially useful, but feedback was that word counts rather than percentages might be more helpful
- Data from id.loc.gov (LCSH or LCNAF) information was widely agreed to be useful and helpful
- Showing the predictions *in context* within the text extracts was seen as helpful
- The overall quality of suggestions provided by the prototypes was given mixed review
- The approach to the user interface and general information architecture, visual design, layout and user experience was scored highly by participants across both prototypes

Review


Select
Test
Review
ria
s
tion
Prototype
Test and Refine
Make Selections
Identify and document candidates
Assess
Opportunities
Risks

Performance

- Metrics for identifying metadata fields in e-book texts generally were as expected
- Identifiers such as ISBN and LCCN were identified reliably and accurately
- Subject headings were a challenge given the number of subject headings and the relatively small training and evaluation dataset
- Other fields such as Title, Personal Names, and Publication information fell between these two, with typical performance hovering around 65-80% in real world evaluation against e-book text data
- Dates scored badly, but this is likely to be something that can be easily improved with better training data
- No models approached the state of the art however, we should probably not expect them to given the relative heterogeneity of the input data

Practicality

- Most of the more commonly used frameworks including Spacy, Hugging Face Transformers, FlairNLP, and others, are:
  - Well documented
  - Regularly updated
  - Flexible
  - Easy to use
- There are no reasons to be concerned, overall, about the ability of these tools to be integrated into data processing pipelines, as part of HITL workflows, or to be used in further experiments
- No reason to be concerned about the cost of training these models or running in production
- Generative AI approaches such as ChatGPT and others have much less mature ecosystems
  - Integration with existing workflows is via toolchains that are relatively new, lightly documented, or in flux
  - Quantitative evaluation against a wider range of data potentially needs exploration and additional testing and development

Can we do better?

There are prima facie reasons to believe that the quality of outputs from the approaches tested to date are not the best we can do with similar frameworks.

- Larger and more comprehensive training data
- Better quality training data
  - Better rules for programmatically tagging training and test data based on MARC to ensure that a high percentage of valid data is tagged in the e-book text
  - More manual review of annotated training data
  - Instances of longer texts with markup that can be used to validate/test workflows in real-world scenarios.
  - Potentially synthetic data to increase dataset and size and to measure and mitigate against bias
- Explore other token classification approaches that provide better *ranking* of outcomes to reduce the number of false-positives produced
- Leverage more information about document structure to target ML at just those sections of documents likely to contain relevant information
- Test and train on more non-English e-books

Human-In-the-Loop Workflows

- The prototypes tested in this experiment were based on a relatively light-touch analysis of user needs.
- Evaluation of the user facing assisted cataloging prototypes suggests that:
  - Use of authority data was valuable
  - Review of data *in-context* was valuable
- But for other automated description, more research is needed to identify:
  - What data is most useful to catalogers
  - How that data should be best presented to catalogers
- More work is needed to iterate towards producing *full* bibliographic records as valid MARC (or BIBFRAME) via automated methods

Towards Piloting Computational Description has just kicked-off (end of September 2023) and has answering these questions as part of the remit.

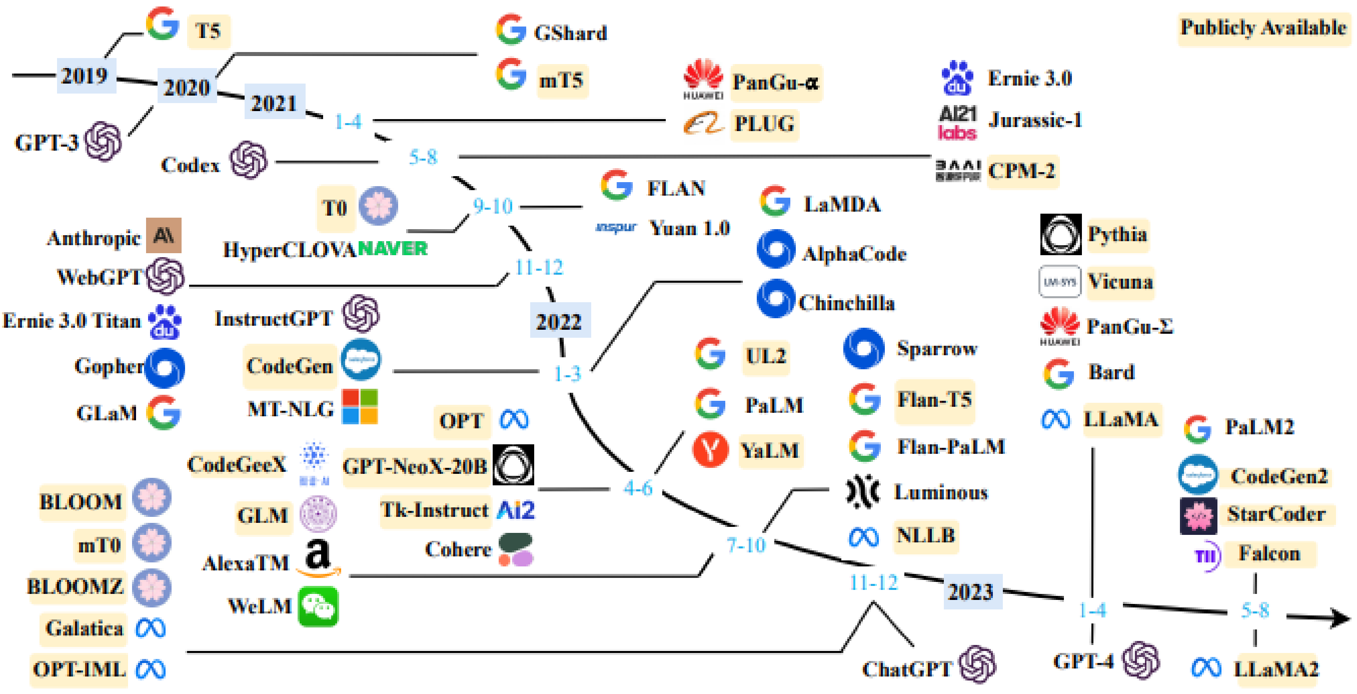


Figure above from [2303.18223] A Survey of Large Language Models

Landscape

- The landscape of AI/ML is completely in flux at the moment.
- New frameworks and models are coming online all the time.
- These models and frameworks often have very basic, and immature, tooling for:
  - Integration with other workflows
  - Processing data at scale
  - Handling longer texts
  - Evaluation
- It is not clear what the overall costs and performance of generative AI models might be
  - Compute cost for self-hosted solutions can be prohibitive
  - Per use costs for commercial APIs can be opaque
- Generative AI models may contain the data we are evaluating against *in their training dataset*
- Tooling or approaches that work for one generative AI model may not work for another, or even for a different version of the same AI model
- There are good prima facie reasons to believe there is a lot of potential for generating high quality bibliographic metadata using LLMs but reasons to be wary

Committing to Automated Description

Should the Library commit to automated description?

- ***Policy Question***: are the measured quality standards for data outputs from automated description ***acceptable*** to the Library?
- ***Stability***: Is the landscape of current AI/ML development stable enough to be sure that investing and fully committing to automated description is the right thing to do *now?*
- ***Information***: Does the Library have a clear enough understanding, especially vis a vis the bleeding edge, of:
  - Costs
  - Performance
  - Risks

Committing to Automated Description

This does not have to be an all-or-nothing decision.

- ***Incremental implementation***: Could partially commit by adopting some automated methods as part of HITL workflows for specific metadata fields where quality is high enough
- ***Engage and review***: Can commit to regular review and assessment of the current state of the art with a view to future decisions to commit more fully to investment in automated description

Staying Current

- Create a framework that allows for regular evaluation and assessment of the state of the art
    - Curated training data and test data for e-books and other relevant datasets
    - Adoption of standard metrics for:
        - Subject Headings and Genre classification
        - Other metadata fields: Title, Personal Names, Publication information, Unique Identifiers, etc
    - Rule-driven mappings between datasets and MARCXML or BIBFRAME to facilitate regular review
    - Tooling for manual review, and for reporting to decision makers and policy holders
- Continue a program of regular engagement with the state of the art specifically as it interfaces with opportunities in automated description
- Deepen understanding of the ways in which automated description can be of immediate practical use in the workflows of expert catalogers and other library users and stakeholders

Thank you!