Literature Review

Paper Name	Link	Date	Notes
CiteSeer: An Automatic Citation Indexing System	https://dl.acm.org/doi/pdf/10.1145/276675.276685	1998	
GIANT: The 1-Billion Annotated Synthetic Bibliographic-Reference-String Dataset for Deep Citation Parsing	https://ceur-ws.org/Vol-2563/aics_25.pdf	2019	
Neural ParsCit: a deep learning-based reference string parser	https://link.springer.com/article/10.1007/s00799-018-0242-1	2018	
ParsCit: An open-source CRF reference string parsing package	https://www.cs.brandeis.edu/~marc/misc/proceedings/lrec-2008/pdf/166_paper.pdf	2008	
Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers	https://dl.acm.org/doi/pdf/10.1145/3197026.3197048	2018	
Deep Reference Mining From Scholarly Literature in the Arts and Humanities	https://www.frontiersin.org/journals/research-metrics-and-analytics/articles/10.3389/frma.2018.00021/full?source=post_page----------	2018	
A New Dataset for Fine-Grained Citation Field Extraction	https://openreview.net/pdf?id=ffO1Piqs1KZo5	2013	
GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications	https://link.springer.com/chapter/10.1007/978-3-642-04346-8_62	2009	No access
Citation segmentation from sparse & noisy data: A joint inference approach with Markov logic networks	https://academic.oup.com/dsh/article-abstract/31/2/333/2462496?login=true#no-access-message	2016	No access, tackles the problem of multilanguage.
Annotated References in the Historiography on Venice: 19th–21st centuries	https://account.openhumanitiesdata.metajnl.com/index.php/up-j-johd/article/view/johd.9	2017	

Bidirectional LSTM-CRF models for sequence tagging	https://arxiv.org/abs/1508.01991	2015	
Attending to Characters in Neural Sequence Labeling Models	https://arxiv.org/abs/1611.04361	2016	
Deep Active Learning for Named Entity Recognition	https://arxiv.org/abs/1707.05928	2017	
Enhancing bibliographic reference parsing with contrastive learning and prompt learning	https://www.sciencedirect.com/science/article/pii/S0952197624007061	2024	
TransParsCit: A Transformer-Based Citation Parser Trained on Large-Scale Synthesized Data	https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1133&context=computerscience_etds	2022	
Neural Architecture Comparison for Bibliographic Reference Segmentation: An Empirical Study	https://www.mdpi.com/2306-5729/9/5/71	2024	
Synthetic vs. Real Reference Strings for Citation Parsing, and the Importance of Re-training and Out-Of-Sample Data for Meaningful Evaluations: Experiments with GROBID, GIANT and Cora	https://arxiv.org/abs/2004.10410	2020	
Comparing free reference extraction pipelines	https://link.springer.com/article/10.1007/s00799-024-00404-6	2024	

CiteSeer: An Automatic Citation Indexing System

It talks about a tool that was developed to help know where a paper has been used (referenced) and what papers it references and provide the context of the citations of the paper.
I didn't finish reading the paper since it wasn't relevant to the topic I was looking for, but it was a nice/old paper about coupling the paper and the context it was referenced in inside other papers.

GIANT: The 1-Billion Annotated Synthetic Bibliographic-Reference-String Dataset for Deep Citation Parsing

This paper discusses the idea of not having enough datasets to train a deep learning model on the parsing of bibliographical references, it tackles the problem by using crossref to collect various data from different fields, topics, and types and then generate (re-parse) the data using a Citation Style Language (CSL) Processor Citeproc-js to modify the reference and generate it in almost 1500 styles to enrich the dataset and can train on different DL model.
The paper also makes a comparison of different DL models which were trained on small datasets but should have a promising output.

Neural ParsCit: a deep learning-based reference string parser\

This paper uses a Neural ParsCit and compares its performance to the original CRF-based ParsCit model, the model does give better results than the CRF-based model and it uses the following:

- Trained word embedding for the problem using pre-trained embeddings (Word2Vec), but training the embedding was a little distracting.
- Character level embeddings and concatenating it with the word embeddings to form the input for the input for the neural model.
- BLSTM model topped with a CRF, because they noticed that BLSTM is good with long-range references but misses in the short-distance labeling so the use of CRF helped improve the performance of the model

The model tackles the problem of multi-lang references by translating them into English labeling them and then applying the labels to the original reference.
I think this paper shows promise and we could use many of its ideas in developing the model we're looking for.

ParsCit: An open-source CRF reference string parsing package

This paper talks about developing ParsCit which is a reference parsing tool, it uses the CRF model to work on the task and it uses the CORA which is a famous dataset regarding this task.

This paper also explains the features extracted from the token and how each token is represented by a feature as we did in the practice project when imitating the AnyStyle tool. It has some different features but they are mainly the same.
It also works on segmenting the references from a paper provided as a text document and then parsing the references later on (Which is a task I don't think we need to do since we're only interested in reference parsing)
One of the early approaches they used was using referencing in multiple rounds:

- Parsing the reference
- Global round that checks neighboring references in case it is helpful to check the referencing style and enhance it

This could help parse multiple references, but I don't think it'll be useful.

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

This paper compares the tools and data that were provided to them for chemical parsing.
They started the comparison without retraining the models and found out that the ML-based model is better than non-ML-models (Rules, templates, regular expressions).
After that, they re-trained the best three ML models (GROBID, CERMINE, and ParsCit) according to the results of their data from previous testing, and found out that there was an enhancement in the results.
This paper also mentions the Neural ParsCit that was reviewed earlier as the only deep learning model they found to that date (2018), and they couldn't test it because they weren't able to install it due to errors.

Deep Reference Mining From Scholarly Literature in the Arts and Humanities

This paper offers a very detailed explanation of an architecture for being able to detect a reference inside a text, and detect it in one of these three tasks:

1. Task 1: reference components. Each token is classified using a taxonomy of 27 specific tags.
2. Task 2: reference typology. Each token is classified according to the generic annotation scheme. As mentioned above, tags include: primary sources (e.g., archival documents), secondary sources (books), and meta-sources, i.e., publications contained within other publications (e.g., journal articles)
3. Task 3: reference span. Each token is classified simply using the begin, end, in, and out schema.

They used a multi-task model to be able to do these three tasks on a text, and in our case, we are only interested in the first task since we're only parsing references.

The baseline model used here is a CRF model they trained on the data set that they have, the data set contains text from humanities and art papers (publications on the historiography on Venice).
They used word embeddings (pre-trained and tuned according to the data set), character-level word embedding (trained on the available dataset), and BiLSTM and CRF for the model.
The architecture is similar to the one used in Neural ParsCite, and they only referenced Neural ParsCite once as they said it wasn't published completely (Neural ParsCite May 2018, and this one July 2018)
The paper tested multiple hyperparameters using grid-search and the best results were chosen, I guess this paper also offers a good advantage in designing the model we're gonna work on.

A New Dataset for Fine-Grained Citation Field Extraction

The paper discusses providing a new dataset different from the CORA dataset that was more variant and had a more detailed segmentation of references.
I read this because it was mentioned in one of the papers I have read before, it is quite old and in the future work section, they are working on expanding it. However, the way the references were segmented didn't feel aligned with the titles that are used in other data/models.
It's good to see if there is future development in this regard and check if there is a bigger dataset by now.

Annotated References in the Historiography on Venice: 19th–21st centuries

The paper offers a new dataset that was manually annotated to help the models learn more from the available data.
The paper trains two models to not only identify the specific classes but also to give a general classification for the references just like task 2 in the "Deep Reference Mining From Scholarly Literature in the Arts and Humanities" paper. And according to the paper, we can use these models as a baseline for our training.
The downside of this data is that it is field-specific, but it has data with many languages and different styles (from the 19th to the 21st).

Bidirectional LSTM-CRF models for sequence tagging

This paper uses sequence labeling tasks (POS, chunking, NER) to train different models(CRF, LSTM, LSTM-CRF, BI-LSTM, BI-LSTM-CRF) to test the results on them and compare them.
According to their training methods, they achieved good scores but they weirdly used Senna embedding, and while comparing with other models they said that it might be the embeddings that gave others better results.

Not only does the system use embedding but the also uses features (word, spelling, context) for the input by connecting them to the output layer, not the input of the model, which could be a good idea to use in our reference labeling task.

Attending to Characters in Neural Sequence Labeling Models

This paper proposes using character-level embeddings to train a sequence labeling model.
It also discusses the different options of using only word embeddings, concatenate word embedding character-level embedding, and using a model to choose different features from both word embeddings and character embeddings)
The model uses Word2Vec for the word embedding and fine-tunes it according to the data randomly initializes the character embeddings and trains them.
The paper says that it should good results enhanced from compared models in these sequence labeling tasks (NER, POS, chunking, and Error det)
And it also uses a BI-LSTM with CRF.

Deep Active Learning for Named Entity Recognition

This paper talks about training a model for NER tasks, it explores a new model that compares to other models' architecture. The paper suggests using CNN-CNN-LSTM architecture which uses the CNN for character and word encoding and an LSTM for decoding.
The paper admits that LSTM encoders are better, but they're slightly better and much slower than CNN encoders which is why this paper compares time between models.
Also, this paper explores using Active Learning which says that models trained using active learning achieve similar results with less amounts of data.

Enhancing bibliographic reference parsing with contrastive learning and prompt learning

This paper introduces a new approach that contains contrastive and prompt learning to do the reference parsing task. In the paper, they developed a dataset containing Chinese and English references. Still, they trained each model alone (as I understood) and then compared results with other models/methods that they trained too (BiLSTM + CRF, CNN, BERT, BERT+ BiLSTM + CRF) and according to their results, their model outperformed these models.
It is a good model but I think that the labels that are used during training don't cover a lot of the data in the real world, that's why they used their data.
A weird observation is that usually, the other models in other papers might perform well in detecting different segments of the references but always have a gap between the title and container-title/journal because they always fail to distinct between them, here there was a gap not as much as I saw in other papers.

TransParsCit: A Transformer-Based Citation Parser Trained on Large-Scale Synthesized Data

This paper uses the dataset that I have read before which has a lot of data in different styles, according to the paper/thesis it uses a Transformer with CRF to do the task of reference-parsing.
It uses Word2Vec as a word embedding and it concatenates it with character embedding as seen in different papers.
The paper compares the model with Neural ParsCit and it says that it couldn't achieve better results but only because they were comparing on one dataset which was CORA (and Neural ParsCit is trained on it - as I recall) while the TransParsCit is trained on data from different fields and different styles GIANT (not only computer science references)
The paper deleted multiple tags and specified some of them. And it doesn't work on multilanguage references.

Neural Architecture Comparison for Bibliographic Reference Segmentation: An Empirical Study

This study has almost done it all, they have built three models trained/tested on the GIANT dataset and then tested it on the CORA dataset. The three models that were built are:

1. CRF
2. BiLSTM + CRF
3. Transformer (encoder) + CRF

The BiLSTM model achieved the best results, and according to the paper, it did this in the testing phase: "The BiLSTM + CRF model demonstrated high performance, achieving nearly perfect scores with an F-score of 0.9998, a Precision of 0.9997, and a Recall of 0.9999." and then they tested and compared the models on the CORA dataset and these were the results: "F-score of 0.9612, a precision of 0.9653, and a recall of 0.9572", I have only showed the BiLSTM results since it was the superior model out of them all.
The thing that they didn't mention is the multilanguage thing which I think we could dig deeper into that part, but still, that needs datasets and more working on.
And this study used Byte-Pair embedding for words.

Synthetic vs. Real Reference Strings for Citation Parsing, and the Importance of Re-training and Out-Of-Sample Data for Meaningful Evaluations: Experiments with GROBID, GIANT, and Cora

This paper compared two CRF models on synthetic data and real data and the results are the same, both models showed similar results when trained on real data and tested on the same dataset or vice versa.
Both models were tested on an outer test set and showed similar results too.

Additionally, the GIANT model was trained on different sizes of datasets and showed enhancement when increasing the size until 10000, and after the scores stopped increasing and in some cases, they started declining.
Also, the paper shows that having more properties in the references could help in increasing the accuracy of the model.

Comparing free reference extraction pipelines

I started reading this paper to see the comparison between different models but unfortunately, the models that are used for comparison are CRF-based, like AnyStyle or Grobid.
I kept reading a little to know what datasets they were using in case it was useful in the project but they weren't using any big datasets or known ones and they were maybe PDF datasets that had scanned files (By OCR) and the comparison of the to extract them and segment the references.
I haven't read it fully but I can read later in case we need more ideas from it, or it was mentioned in future papers.