data

MDPI

Article

Neural Architecture Comparison for Bibliographic Reference Segmentation: An Empirical Study

Rodrigo Cuéllar Hidalgo [1], Raúl Pinto Elías [2], Juan-Manuel Torres-Moreno [3,*], Osslan Osiris Vergara Villegas [4], Gerardo Reyes Salgado [5] and Andrea Magadán Salazar [2]

1 Biblioteca Daniel Cosío Villegas, El Colegio de México, Carretera Picacho Ajusco 20, Mexico City 14110, Mexico; rcuellar@colmex.mx
2 Tecnológico Nacional de México/CENIDET, Cuernavaca 62490, Mexico; raul.pe@cenidet.tecnm.mx (R.P.E.); andrea.ms@cenidet.tecnm.mx (A.M.S.)
3 Laboratoire Informatique d'Avignon, Université d'Avignon, 339 Chemin des Meinajariès, CEDEX 9, 84911 Avignon, France
4 Industrial and Manufacturing Engineering Department, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez 32310, Mexico; overgara@uacj.mx
5 Departamento de Informática y Estadística, Universidad Rey Juan Carlos, Av. del Alcalde de Móstoles, 28933 Madrid, Spain; gerardo.reyes@urjc.es
* Correspondence: juan-manuel.torres@univ-avignon.fr

Abstract: In the realm of digital libraries, efficiently managing and accessing scientific publications necessitates automated bibliographic reference segmentation. This study addresses the challenge of accurately segmenting bibliographic references, a task complicated by the varied formats and styles of references. Focusing on the empirical evaluation of Conditional Random Fields (CRF), Bidirectional Long Short-Term Memory with CRF (BiLSTM + CRF), and Transformer Encoder with CRF (Transformer + CRF) architectures, this research employs Byte Pair Encoding and Character Embeddings for vector representation. The models underwent training on the extensive Giant corpus and subsequent evaluation on the Cora Corpus to ensure a balanced and rigorous comparison, maintaining uniformity across embedding layers, normalization techniques, and Dropout strategies. Results indicate that the BiLSTM + CRF architecture outperforms its counterparts by adeptly handling the syntactic structures prevalent in bibliographic data, achieving an F1-Score of 0.96. This outcome highlights the necessity of aligning model architecture with the specific syntactic demands of bibliographic reference segmentation tasks. Consequently, the study establishes the BiLSTM + CRF model as a superior approach within the current state-of-the-art, offering a robust solution for the challenges faced in digital library management and scholarly communication.

Keywords: reference mining; BiLSTM; transformers; byte-pair encoding; Conditional Random Fields

Citation: Cuéllar Hidalgo, R.; Pinto Elías, R.; Torres-Moreno, J.-M.; Vergara Villegas, O.O.; Reyes Salgado, G.; Magadán Salazar, A. Neural Architecture Comparison for Bibliographic Reference Segmentation: An Empirical Study. *Data* **2024**, *9*, 71. https://doi.org/10.3390/data9050071

Academic Editors: Filipe Portela, Edson Talamini and Letícia De Oliveira

Received: 14 March 2024
Revised: 12 May 2024
Accepted: 15 May 2024
Published: 18 May 2024

1. Introduction

In recent years, there has been a significant growth in electronic scientific publications, driven by technological advances, the rapid expansion of the World Wide Web (WWW), and largely many of these publications emerge directly in digital format, considerably accelerating their availability [1–3].

Digital libraries have become crucial resources for scientific and academic communities, serving not just as repositories for publications but also as platforms for information classification and analysis. This enhances the ability to group and retrieve relevant data. Accurate recording and analysis of citations and bibliographic references are particularly important.

In the digital era, the surge in scientific publications has necessitated advanced solutions for managing and processing large volumes of bibliographic data. As libraries and information repositories move towards comprehensive digitization, the need for efficient and accurate bibliographic reference segmentation has become paramount.

The recording and analysis of citations and bibliographic references not only allow measuring the impact of a publication in the scientific field but also extracting valuable information, such as the disciplines citing a specific work, the geographic regions where it is most read, or the language in which it is most cited. This enables libraries to identify needs and opportunities in their activities of material acquisition and building special collections [4].

Given the exponential growth of scientific and academic literature, automated processes for tasks such as storage, consultation, classification, and information analysis become essential. The first step to achieve this, is the correct detection, extraction, and segmentation of bibliographic references (also known as "reference mining") within academic documents [5].

In the literature, ref. [6] conducted a comparative study among different approaches to citation extraction, including Conditional Random Fields (CRF), regular expressions, rules, template matching, and LSTM neural networks. A key aspect of their methodology was maintaining uniformity in the dataset used for training, ensuring that the only variable was the extraction technique itself. However, limitations in the availability and operational functionality of some tools, particularly those based on LSTM networks, prevented a comprehensive evaluation of all approaches as the code for LSTM models was not available. Despite these constraints, their assessment found that CRF-based implementations performed best on the specially prepared dataset. Complementing this analysis, a study in [7] compares datasets containing real and synthetic bibliographic references, concluding that both types are suitable for training reference segmentation models. After retraining models from these tools, CRF approaches not only outperformed others in precision but also demonstrated significant adaptability to various extraction requirements and citation styles. These findings underscore the challenges of testing code from different approaches, which often is not well-documented or is incomplete, and highlight the importance of evaluating, under uniform conditions, the most representative architectures for bibliographic reference segmentation as proposed in this study.

In this paper, we address the task of bibliographic reference segmentation by comparing models based on three distinct natural language processing architectures: Conditional Random Fields (CRF), Bidirectional Long Short-Term Memory with CRF (BiLSTM + CRF), and Transformer Encoder with CRF (Transformer + CRF). These models are evaluated using the Giant corpus for training and the Cora corpus for further assessment, highlighting the capabilities and differences of each architecture in handling the complexities of bibliographic data.

The problem of bibliographic reference segmentation has been tackled using various approaches, ranging from heuristic methods to machine learning (ML) and deep learning (DL) techniques. Conditional Random Fields (CRF) stand out as the most prominent representative of ML approaches. However, DL-based approaches exhibit notable variability, as they employ diverse types of embeddings and context-capturing architectures such as LSTM or Transformers [8,9]. The purpose of this study is to evaluate, under uniform conditions, the three most representative architectures for segmenting bibliographic references. Despite advances in natural language processing techniques, bibliographic reference segmentation continues to present unique challenges, especially due to the variety of formats and styles, as well as the presence of specialized terminology and proper names. In this context, our study focuses on identifying the most effective architecture for this task, considering factors such as accuracy and efficiency.

We present a comparative evaluation of three natural language processing architectures and an analysis under uniform conditions, emphasizing the BiLSTM + CRF model's superiority. This model's ability to handle complex syntactic loads highlights the importance of selecting architecture based on specific task demands, contributing valuable insights for digital library management and automated bibliographic reference processing.

The rest of the paper is organized as follows: Section 2 presents an overview of the various approaches that have been used to address reference segmentation. Section 3 not only

describes the data sets used but also details the preprocessing steps applied to prepare the data for model training. In Section 4, we detail the implemented architectures, each encapsulated in a different model, and their respective training processes and evaluation. Section 5 addresses the experimentation carried out and discusses the results. Finally, Section 6 presents the conclusions, highlighting the effectiveness of the BiLSTM + CRF model in comparison with other techniques and discussing the implications of these findings for managing digital libraries and the automated processing of bibliographic references.

2. State-of-the-Art

The challenge of reference segmentation has persisted as an open research problem for decades, with numerous attempts made towards its efficient resolution. Each effort has approached the problem from a unique perspective. Hence, it is crucial to understand the primary function of a citation parser, which is to take an input string (formatted in a specific style such as APA, ISO, Chicago, etc.), extract the metadata, and then generate a labeled output [8].

In 2000, the first attempts to automate the segmentation of bibliographic references emerged [10], focusing on the syntactic analysis (parser) of online documents based on HyperText Markup Language (HTML) and simultaneously proposing the transformation of other formats like Portable Document Format (PDF) to HTML or Extensible HyperText Markup Language (XHTML). Many of the approaches consist of using different proposals for the syntactic analysis of information, using techniques similar to web scraping (a set of techniques that use software programs to extract information from websites), the use of character pattern identification, also known as regular expressions, for their use as labels (for example, the 'pp' associated with the number of pages, etc.) that allow establishing analysis contexts for the identification and extraction of these.

Other works employ machine learning-based models for the syntactic analysis of strings containing references, such as the case of Hidden Markov Models [11]; the clustering proposal through the TOP-K algorithm [12]; or the Conditional Random Fields model, implemented in the GROBID and CERMINE systems [13–15], which seems to be the technique that has given the best results.

From 2018 onwards, works based on deep learning began to emerge, improving the precision of the results obtained with machine learning. These works usually use a Long Short-Term Memory neural network architecture (LSTM), which combines its output with the Conditional Random Fields (CRF) technique [16,17].

It is important to mention the particular case of the ParsRec system, which has the peculiarity of being an approach based on recommendation and meta-learning. The main premise of ParsRec is "Although numerous reference parsers address the problem from different approaches, offers optimal results in all scenarios". ParsRec recommends the best analyzer, out of the ten contemplated, based on the reference string in turn [18].

A detail worth highlighting is the fact that all the main proposals based on Deep Learning [14,16,17] make use of vector representations based on Word2Vec and ELMO.

Lastly, we have the case of Choi et al. [9], who propose a model based on transfer learning using BERT [19] as a base, which its authors claim is the best exponent in the state-of-the-art for working with multilingual references.

3. Datasets and Preprocessing Methodology

In the realm of bibliographic reference segmentation, the choice of an appropriate training dataset is pivotal for the development of models that are both robust and generalizable.

3.1. Bibliographic Data

The Giant corpus [20] was selected as the training dataset for its breadth and diversity in bibliographic references. This comprehensive dataset encompasses a vast array of citation styles and document types, presenting a rich tapestry of bibliographic data that spans across numerous academic disciplines and publication formats. By training models

on such a heterogeneous dataset, we aim to cultivate an architecture that learns the intricate patterns and variations inherent in bibliographic references and possesses the versatility to adapt to the myriad ways academic information can be structured. As for the evaluation of the model's efficiency on an independent dataset, the CORA corpus [21] was used, distinguished by its detailed structure and frequent use as a benchmark in reference segmentation studies [7,15,16].

3.1.1. Giant Corpus

The Giant corpus contains 991,411,110 records, divided into 1568 different citation styles and encompasses 24 types of documents (https://doi.org/10.7910/DVN/LXQXAO, accessed on 15 January 2024) [20]. Each record has the following structure (Listing 1).

Listing 1: Giant record example.

```
{
  "doi": "10.2307/2177340",
  "articleType": 3,
  "citationStyle": 0,
  "citationStringAnnotated": "<author><family>Ritchie</family
     >, <given>E.</given> and <family>Powell</family>, <given>
     Elmer Ellsworth</given></author> (<issued><year>1907</
     year></issued>) <title>Spinoza and Religion.</title> <
     container-title>The Philosophical Review</container-title
     >, <volume>16</volume>(<issue>3</issue>), p. <page>339</
     page>. [online] Available from: <URL>http://dx.doi.org
     /10.2307/2177340</URL>"
}
```

Each field contains the following information:

- **doi**: Digital Object Identifier (DOI) is an unique identifier of the document it represents (https://ask.library.uic.edu/faq/345899, accessed on 17 January 2024).
- **articleType**: The identifier representing the type of document (thesis, article, book, etc.).
- **citationStyle**: The identifier representing the citation style (APA, Harvard, IEEE, etc.).
- **citationStringAnnotated**: The annotated reference string.

The reference string comes annotated with the following eXtensible Markup Language (XML) structure (Listing 2):

Listing 2: Annotated reference string example.

```
<author>
        <family>Ritchie</family>,
        <given>E.</given> and
        <family>Powell</family>,
        <given>Elmer Ellsworth</given>
</author> (
<issued>
        <year>1907</year>
</issued>)
<title>Spinoza and Religion.</title>
<container-title>The Philosophical Review</container-title>,
<volume>16</volume>(
<issue>3</issue>), p.
<page>339</page>. [online] Available from:
<URL>http://dx.doi.org/10.2307/2177340</URL>
```

The "citationStringAnnotated" in the Giant corpus provides an annotated reference string for each record, using XML-like tags to mark different bibliographic elements such

as authors, titles, and publication details. This structured annotation facilitates the precise extraction of bibliographic information, crucial for training models to accurately segment and understand the various components of academic references.

3.1.2. CORA Corpus

It is a human-annotated corpus of 1877 bibliographic reference strings with a variety of formats and styles, including magazine preprints, conference papers, and technical reports. (https://people.cs.umass.edu/ mccallum/data/cora-refs.tar.gz, accessed on 22 January 2024) [21].

3.2. Preprocessing

The necessity to reprocess the data stemmed from a strategic decision aimed at reducing the variability and computational complexity inherent in the original dataset. Given the vast diversity and multitude of citation styles and document types in the Giant corpus, the initial data presented a significant challenge in terms of model training and evaluation. The variety in formatting and structuring of bibliographic references, while valuable for understanding real-world application scenarios, introduced a level of complexity that could potentially hinder the model's ability to learn consistent patterns and generalize across unseen data.

To address this, we simplified the dataset and streamlined the annotation structure, focusing on distilling the essential bibliographic components most relevant for the task of reference segmentation. This preprocessing step was designed to minimize extraneous variability that does not contribute to the core objective of the study. By reducing the number of variables, we aimed to eliminate redundant or non-informative features that could obscure the significant patterns necessary for effective model learning.

Moreover, the selected attributes were those that consistently appear across different citation styles and document types, ensuring that the models focus on learning the fundamental syntactic and structural features of bibliographic references. This not only reduces the computational demands on the models, enhancing their efficiency, but also helps in improving their generalizability. By training the models on a more streamlined set of data, they are better equipped to accurately identify and extract relevant bibliographic information from a wide range of academic documents, reflecting a balance between model complexity and performance efficiency.

Preprocessing the data in this manner not only facilitates a more focused and efficient training process but also significantly enhances the models' ability to perform accurately in real-world scenarios where bibliographic data may vary in presentation but not in fundamental structure. This approach ensures that our models are not only theoretically sound but also practically viable in diverse academic and research settings.

3.2.1. Preprocessing Giant Corpus

For the purposes of this work, the annotated reference was simplified in an automated manner to maintain only the following labels, which are considered the minimum necessary to identify a work (Table 1):

Table 1. Simplified labels

-Author	- Year	-Title	-Container-Title
-Volume	-Issue	-Page	-ISBN
-ISSN	-Publisher	-DOI	-URL

Resulting in the following format (Listing 3):

Listing 3: Annotated references with simplified tags.

```
<author>
Ritchie E. and Powell, Elmer Ellsworth
```

```
</author>
(
<year>
1907
</year>
)

<title>
    Spinoza and Religion.
</title>
<container-title>
    The Philosophical Review
</container-title>,
<volume>
    16
</volume>
(
<issue>
    3
</issue>
), p.
<page>
    339
</page>
. [online] Available from:
<URL>
    http://dx.doi.org/10.2307/2177340
</URL>
```

3.2.2. Preprocessing CORA Corpus

In the case of the CORA corpus, the labels were adjusted to align with those used in the Giant training, ensuring consistency across both datasets. The same preprocessing methodology applied to the Giant corpus was also employed for CORA, aiming to standardize the data and reduce variability and complexity. Additionally, 92 references were discarded from CORA due to encoding errors and label duplication, leaving 1787 of the original 1877 references for evaluation. Preprocessing ensures that both datasets are prepared to function correctly for training and evaluating the models, facilitating a direct and fair comparison of their performance on standardized bibliographic data.

4. Architectures of the Evaluated Models

For the development of each of the three models, the Flair NLP framework (https://flairnlp.github.io/, accessed on 29 December 2023) created by [22] was used for the following reasons:

- Its ability to efficiently integrate and manage different types of embeddings.
- Extensible and modular architecture makes it easy to add additional model-specific layers, such as Word Dropout and Locked Dropout.
- Comprehensive documentation and practical examples available.

In addition to the Flair NLP framework, several other tools and libraries were integral to our implementation:

- PyTorch: Serves as the underlying framework for Flair, enabling dynamic computation graphs and tensor operations for model training and evaluation. PyTorch was also used to develop the Transformer encoder architecture, allowing customization for bibliographic reference segmentation.

- Pandas: Used for data manipulation and analysis, assisting in the organization and formatting of data prior to model training.

Computing Requirements:

- The models were trained and evaluated on a machine equipped with an Intel i9 processor and 64 GB of RAM, which supports the processing of large datasets.
- GPU acceleration was employed to enhance the speed of model training, using a NVIDIA A5000 (NVIDIA Corporation Cuernavaca, Mexico).
- Approximately 1000 GB of SSD storage was allocated for storing both raw and processed datasets, as well as the models' state during different phases of training.

These tools and resources were used to manage large-scale data and perform intensive calculations for training models as presented in this paper.

The models evaluated in this study share a common base architecture, which incorporates Byte Pair Encoding (BPE) and Character Embeddings for vector representation. This strategic combination is adept at capturing both the semantic essence of words and the nuanced characteristics at the character level, an approach that proves crucial for addressing variations and common errors encountered in bibliographic references. These variations and errors primarily manifest as omissions in bibliographic fields and variability in writing styles, such as the inversion of the order of names (where last names and first names may be swapped) and the inconsistent expression of volume numbers (sometimes represented in arabic numerals, other times in roman numerals). By accommodating these peculiarities, the chosen embedding strategies enhance the models' ability to accurately process and segment bibliographic data, reflecting the complexity and diversity inherent in academic references.

In addition to these representation layers, the common architecture of the models includes several additional layers designed to optimize performance and generalization:

- Word Dropout: This layer reduces overfitting by randomly "turning off" (i.e., setting to zero) some word vectors during training, which helps the model not to rely too much on specific words.
- Locked Dropout: Similar to word dropout, but applied uniformly across all dimensions of a word vector at a given step. This improves the robustness of the model by preventing it from overfitting to specific patterns (token combinations) in the training data.
- Embedding2NN: A layer that transforms concatenated embeddings into a representation more appropriate for processing by subsequent layers. This transformation can include non-linear operations to capture more complex relationships in the data.
- Linear: A linear layer that acts as a classifier, mapping the processed representations to the target segmentation labels.

Furthermore, this study tested three models, each incorporating a specific processing layer that capitalizes on the strengths of distinct approaches. These models, set to be detailed in the following subsections, were selected based on their status as the most recently utilized and best-performing architectures in the literature for reference segmentation. The choice of these three architectures allows for an ideal comparison, as they represent the cutting edge in tackling the complexities of bibliographic data [9,14,17], providing a comprehensive overview of current capabilities and identifying potential areas for innovation in reference segmentation techniques.

4.1. CRF Model

The CRF model focuses on using Conditional Random Fields for sequence segmentation [23]. This technique is particularly effective in capturing dependencies and contextual patterns in sequential data (see Figure 1).

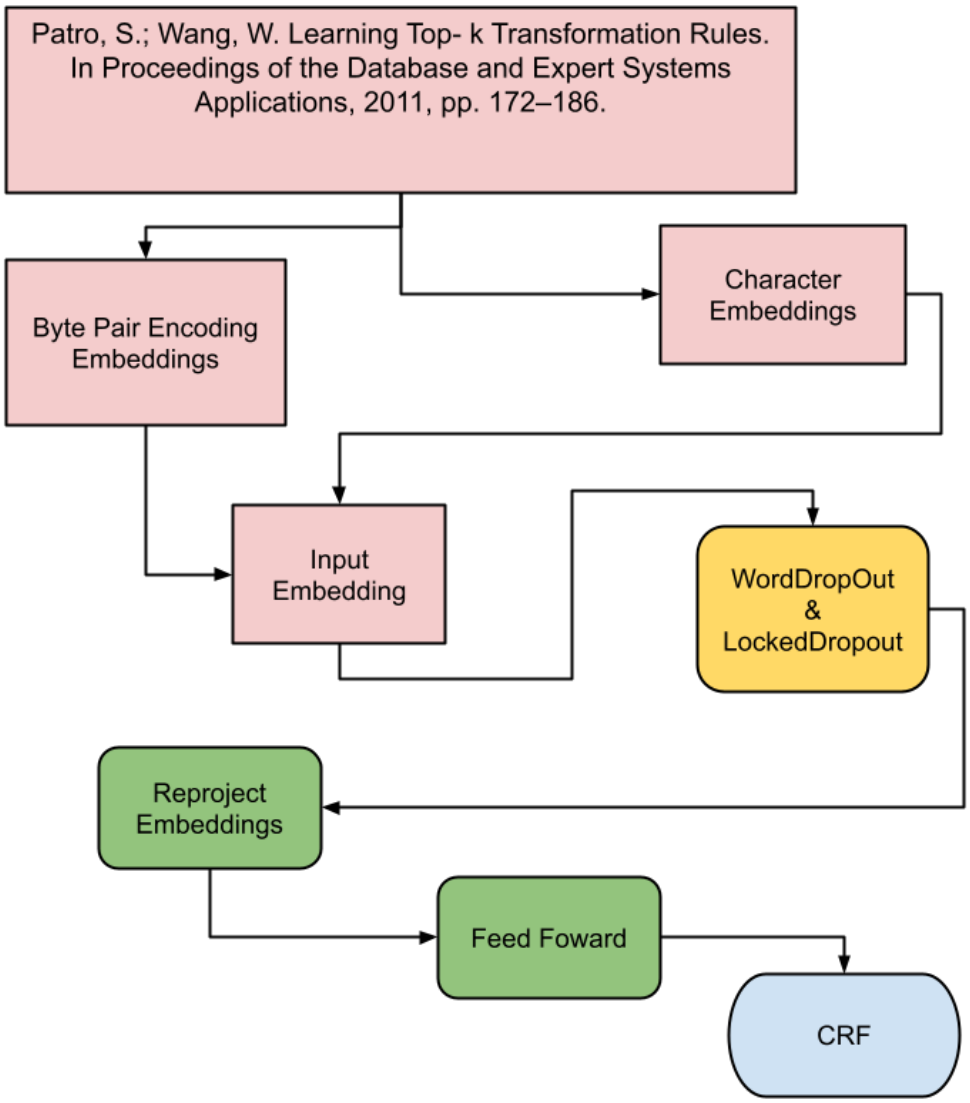

Figure 1. Graphical representation of the CRF model [12].

The following outlines the layers that comprise the architecture of the CRF model (Listing 4):

Listing 4: CRF model.

```
Model: "CRF(
  (embeddings): StackedEmbeddings(
    (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi
       -100000-50)
    (list_embedding_1): CharacterEmbeddings(
      (char_embedding): Embedding(275, 25)
      (char_rnn): LSTM(25, 25, bidirectional=True)
    )
  )
  (word_dropout): WordDropout(p=0.05)
  (locked_dropout): LockedDropout(p=0.5)
  (embedding2nn): Linear(in_features=650, out_features=650,
     bias=True)
```

```
  (linear): Linear(in_features=650, out_features=29, bias=True
     )
  (loss_function): ViterbiLoss()
  (crf): CRF()
)"
```

The description of the CRF model outlines its configuration and structure in terms of components and their functionalities:

Model: "CRF": This line denotes the name of the model, the following vignettes represent the layers that compose it:

- **(embeddings): StackedEmbeddings**:
  Refers to combining various types of word embeddings to form a rich and complex representation. It utilizes BytePairEmbeddings and CharacterEmbeddings:
  - **(list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)**: BytePairEmbeddings are based on Byte Pair Encoding (BPE), capturing subword-level semantics, useful for handling out-of-vocabulary (OOV) words. The specific BPE model used is indicated by "model=0-bpe-multi-100000-50", detailing its parameters.
  - **(list_embedding_1): CharacterEmbeddings**: Uses character-level embeddings, crucial for understanding orthographic peculiarities and common errors in texts.
    - **(char_embedding): Embedding(275, 25)**: Defines a character embedding with a vocabulary size of 275 and 25-dimensional vectors.
    - **(char_rnn): LSTM(25, 25, bidirectional=True)**: A bidirectional LSTM network that processes character embeddings, with 25 units in both directions, capturing contexts before and after each character.
- **(word_dropout): WordDropout(p=0.05)**: Applies dropout at the word level with a probability of 0.05, helping prevent overfitting by randomly "turning off" words during training.
- **(locked_dropout): LockedDropout(p=0.5)**: Applies dropout uniformly across all dimensions of word vectors at a given step, with a probability of 0.5, enhancing the model's robustness.
- **(embedding2nn): Linear(in_features=650, out_features=650, bias=True)**: A linear layer transforming concatenated embeddings representation, preparing them for processing by subsequent layers.
- **(linear): Linear(in_features=650, out_features=29, bias=True)**: Another linear layer acting as a classifier, mapping processed representations to 29 target label categories.
- **(loss_function): ViterbiLoss()**: Employs the Viterbi loss function, suitable for sequential prediction tasks like reference segmentation.
- **(crf): CRF()**: Indicates the use of a Conditional Random Field for sequence label prediction, optimizing the coherence and accuracy of predicted labels.

Each line succinctly summarizes a key component of the CRF model and its role in the learning and prediction process, highlighting the complexity and sophistication of the approach taken for bibliographic reference segmentation. Next, the equations describing the interactions of the components of the CRF model are presented.

embeddings:

$$\text{emb}(w_i) = [\text{embBPE}(w_i); \text{embchar}(w_i)] \tag{1}$$

The embeddings represent the combination of Byte Pair Encoding (BPE) and character embeddings. BPE captures the semantic aspects of words, while character embeddings focus on the syntactic nuances at the character level. This dual approach is crucial for processing bibliographic references, where both semantic context and specific syntactic forms (like abbreviations or special characters) play key roles.

word_dropout:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{emb}(w_i), p = 0.05) \quad (2)$$

The word dropout randomly deactivates a portion of the word embeddings during training (here, 5% as indicated by p=0.05). This method prevents the model from over-relying on particular words, encouraging it to learn more generalized patterns. This approach is particularly beneficial in bibliographic texts where certain terms, such as common author names or publication titles, might appear with significantly higher frequency than others, potentially skewing the model's learning focus.

locked_dropout:

$$\text{emblocked}(w_i) = \text{LockedDropout}(\text{embdropout}(w_i), p = 0.5) \quad (3)$$

Locked dropout extends the dropout concept to entire embedding vectors, turning off the same set of neurons for the entire sequence. This approach helps in maintaining consistency in the representation of words across different contexts, an essential factor in processing structured bibliographic data.

embedding2nn:

$$\text{embnn}(w_i) = \text{Linear}(\text{emblocked}(w_i), 650) \quad (4)$$

This linear transformation adapts the embeddings for further processing by the neural network layers. It is a crucial step for converting the rich, but potentially unwieldy, embedding information into a more suitable format for the classification tasks ahead.

linear:

$$\text{output}(w_i) = \text{Linear}(\text{emb}_{\text{nn}}(w_i), 29) \quad (5)$$

The final linear layer maps the transformed embeddings to the target classes. In this model, there are 29 classes, likely corresponding to different components of a bibliographic reference (like author, title, year, etc.). This layer is pivotal for the actual task of reference segmentation.

crf:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'i-1,y'i} + \sum i = 1^{n} Si, y'_i)} \quad (6)$$

$$Y^* = \arg\max_{Y} P(Y|H) \quad (7)$$

The CRF layer is key for capturing the dependencies between tags in the sequence. It considers not only the individual likelihood of each tag but also how likely they are in the context of neighboring tags. This sequential aspect is vital for bibliographic references, where the order and context of elements (like the sequence of authors or the structure of a citation) are crucial for accurate segmentation.

4.2. BiLSTM + CRF Model

The BiLSTM + CRF model combines Bidirectional Long Short-Term Memory (BiLSTM) networks with CRF [24] to better capturing both past and future context in the text sequence. BiLSTMs process the sequence in both directions, offering a deeper understanding of the syntactic structure (see Figure 2).

The following outlines the layers that comprise the architecture of the BiLSTM + CRF model (Listing 5):

Listing 5: BiLSTM + CRF model.

```
Model: "BiLSTM + CRF(
  (embeddings): StackedEmbeddings(
    (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi
       -100000-50)
```

```
    (list_embedding_1): CharacterEmbeddings(
      (char_embedding): Embedding(275, 25)
      (char_rnn): LSTM(25, 25, bidirectional=True)
    )
  )
  (word_dropout): WordDropout(p=0.05)
  (locked_dropout): LockedDropout(p=0.5)
  (embedding2nn): Linear(in_features=650, out_features=650,
     bias=True)
  (rnn): LSTM(650, 256, batch_first=True, bidirectional=True)
  (linear): Linear(in_features=512, out_features=29, bias=True
     )
  (loss_function): ViterbiLoss()
  (crf): CRF()
)"
```

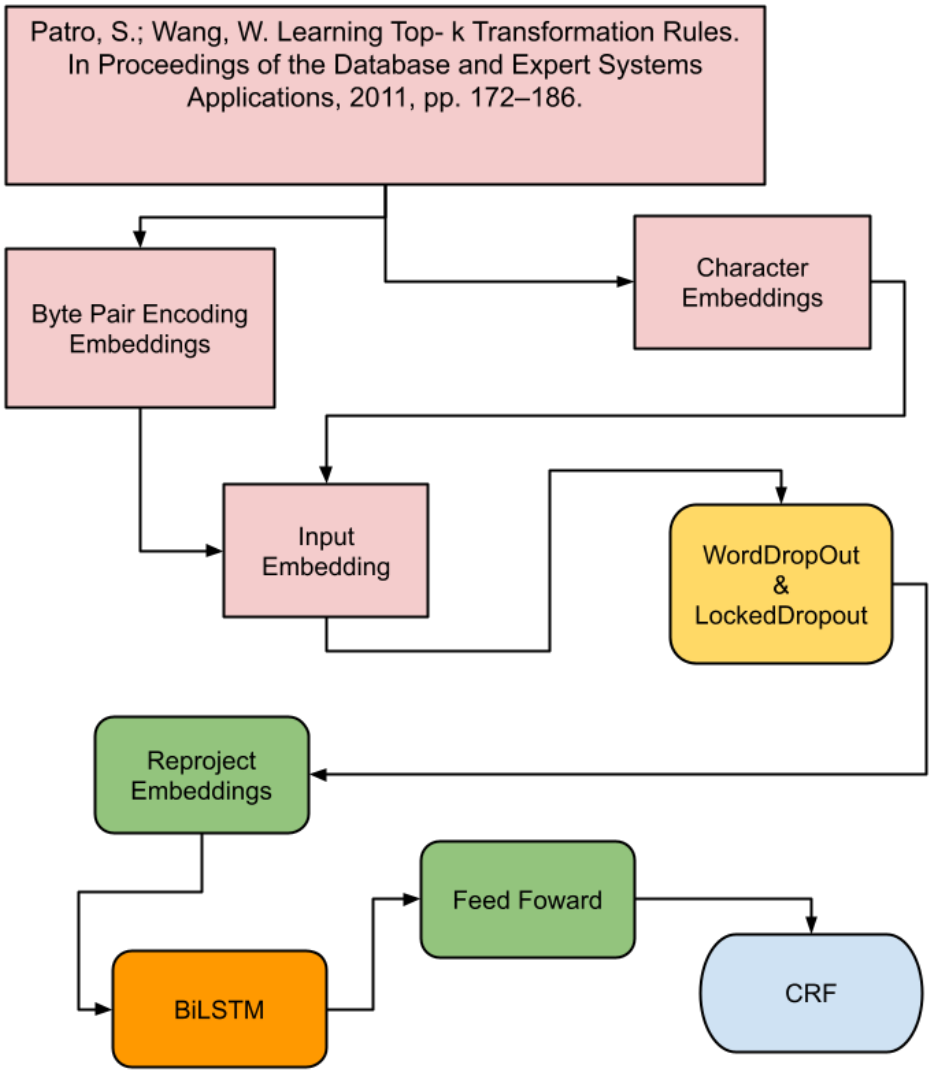

Figure 2. Graphical representation of the BiLSTM + CRF model [12].

This model is fundamentally similar to the CRF, with a key distinction being the incorporation of a Bidirectional Long Short-Term Memory (highlighted in yellow as the **rnn**layer). This layer is strategically positioned between **embedding2nn** and the **linear**. The BiLSTM layer is crucial for capturing both past and future context, which is particularly

beneficial for structured tasks like bibliographic reference segmentation.

The **rnn** layer is described below:

- **(rnn): LSTM(650, 256, batch_first=True, bidirectional=True)**: A bidirectional LSTM layer that processes sequences in both forward and backward directions. With an input size of 650 features and an output of 256 features, it captures contextual information from both past and future data points within a sequence, enhancing the model's ability to understand complex dependencies in bibliographic reference segmentation.

  Let's delve into the mathematical aspects of this layer:

  rnn:

$$\overrightarrow{h_i} = \text{BiLSTMforward}(\text{embnn}(w_i), \overrightarrow{h_{i-1}}) \tag{8}$$

$$\overleftarrow{h_i} = \text{BiLSTMbackward}(\text{embnn}(w_i), \overleftarrow{h_{i+1}}) \tag{9}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{10}$$

These equations represent the forward and backward passes of the BiLSTM. The forward pass $\overrightarrow{h_i}$ processes the sequence from start to end, capturing the context up to the current word. Conversely, the backward pass $\overleftarrow{h_i}$ processes the sequence in reverse, capturing the context from the end to the current word. The final hidden state h_i is a concatenation of these two passes, providing a comprehensive view of the context surrounding each word.

This bidirectional context is invaluable for bibliographic data. For instance, in a list of authors, the context of surrounding names helps in accurately identifying the start and end of each author's name. Similarly, for titles or journal names, the BiLSTM can effectively use the context to delineate these components accurately.

4.3. Transformer + CRF Model

Finally, the model incorporating the Transformer Encoder [25] with CRF leverages the architecture of transformers for global attention processing of the sequence. This approach allows capturing complex and non-linear relationships in the data (see Figure 3).

The following outlines the layers that comprise the architecture of the Transformer + CRF model (listing 6):

Listing 6: Transformer + CRF model.

```
Model: "Transformer + CRF(
  (embeddings): CustomStackedEmbeddings(
    (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi
       -100000-50)
    (list_embedding_1): CharacterEmbeddings(
      (char_embedding): Embedding(275, 25)
      (char_rnn): LSTM(25, 25, bidirectional=True)
    )
(positional_encoding): PositionalEncoding(
 (dropout): Dropout(p=0.1, inplace=False)
)
(transformer_encoder_layer): CustomTransformerEncoderLayer(
 (self_attn): MultiheadAttention(
 (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
 out_features=256, bias=True)
 )
 (linear1): Linear(in_features=256, out_features=512, bias=True)
```

```
  (dropout): Dropout(p=0.1, inplace=False)
  (linear2): Linear(in_features=512, out_features=256, bias=True)
  (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (dropout1): Dropout(p=0.1, inplace=False)
  (dropout2): Dropout(p=0.1, inplace=False)
  (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
  track_running_stats=True)
)
(transformer_encoder): TransformerEncoder(
  (layers): ModuleList(
  (0): CustomTransformerEncoderLayer(
  (self_attn): MultiheadAttention(
  (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
  out_features=256, bias=True)
  )
  (linear1): Linear(in_features=256, out_features=512, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (linear2): Linear(in_features=512, out_features=256, bias=True)
  (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (dropout1): Dropout(p=0.1, inplace=False)
  (dropout2): Dropout(p=0.1, inplace=False)
  (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
  track_running_stats=True)
  )
  )
)
  )
  (word_dropout): WordDropout(p=0.05)
  (locked_dropout): LockedDropout(p=0.5)
  (embedding2nn): Linear(in_features=650, out_features=650,
     bias=True)
  (linear): Linear(in_features=650, out_features=29, bias=True
     )
  (loss_function): ViterbiLoss()
  (crf): CRF()
)"
```

The **positional_enconder**, **transformer_encoder_layer** and **transformer_encoder** layers are described below:

- **(positional_encoding): PositionalEncoding(dropout=0.1, inplace=False)**: This layer adds positional information to the input embeddings, allowing the model to capture the sequence of the text. The addition of positional encodings is crucial for attention-based models, such as transformers, as it enables them to distinguish the order of words in a sequence. The use of *dropout* with a probability of 0.1 helps to prevent overfitting by randomly "turning off" parts of the positional embeddings during training to enhance the model's robustness.

- **(transformer_encoder_layer): CustomTransformerEncoderLayer(...) and (transformer _encoder): TransformerEncoder(...)**: These two layers represent the heart of the Transformer model, where the first defines the structure of a single layer of the transformer encoder, including multi-head attention, residual connections, and layer normalization, while the second stacks multiple of these layers to construct the complete encoder. The apparent redundancy between these layers is because the first specifies the architecture and configuration of an individual layer within the encoder, including specific operations such as attention and normalization, and the second encapsulates the repetition of these layers to form the complete encoder, allowing the model to process and learn from sequences with greater depth and complexity. The inclusion of *BatchNorm1d* in the custom layer suggests an adaptation to stabilize and accelerate training by normalizing activations across mini-batches, which is not typical in standard transformers but can offer benefits in terms of convergence and performance in specific tasks like bibliographic reference segmentation.

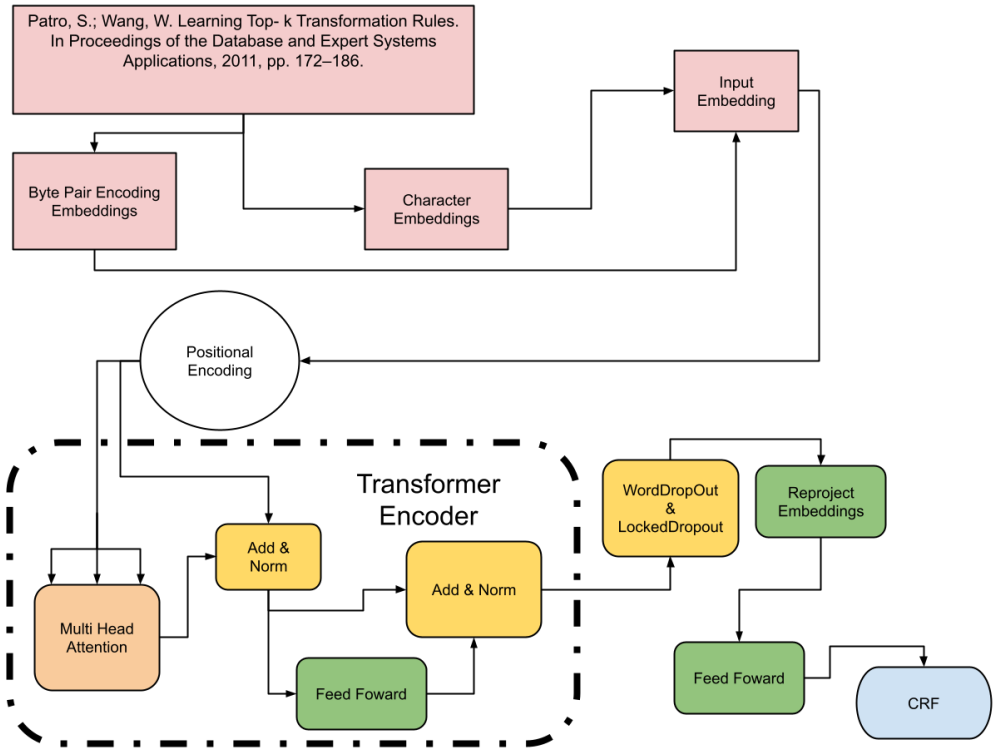

Figure 3. Graphical representation of the Transformer + CRF model [12].

It should be mentioned that in the case of the Transformer + CRF model, placing the Transformer Encoder layer before the embedding2nn layer is due to the following reasons.

First, in bibliographic references, context is of vital importance. The position of a word or phrase can significantly alter its interpretation, such as distinguishing between an author's name and the title of a work. By processing the embeddings through the positional encoder and the Transformer from the beginning, the model can more effectively capture the contextual and structural relationships specific to bibliographic references.

Second, the early inclusion of the Transformer allows for early capture of these contextual relationships. This is crucial in bibliographic references, where the structure and order of elements (authors, title, publication year, etc.) follow patterns that can be complex and varied. The Transformer, known for its ability to handle long-distance dependencies, is ideal for detecting and learning these patterns.

Finally, once contextualized representations are generated, the embedding2nn layer acts as a fine-tuner, making specific adjustments and improvements to these representations. This makes the representations even more suitable for the Named Entity Recognition (NER) task, optimally adapting them for the accurate identification of the different components within bibliographic references.

The model being analyzed here, while structurally similar to the CRF model, introduces a significant variation with the addition of **positional_encoding** and a **transformer_encoder_layer** between the **embedding** and **word_dropout** layers. This inclusion is a key differentiator, enhancing the model's ability to process and understand the sequence data more effectively. Let's explore these additional layers:

positional_encoding

$$\text{emb}_{\text{pos}}(w_i) = \text{PositionalEncoding}(\text{emb}(w_i)) \tag{11}$$

Positional encoding is added to the embeddings to provide context about the position of each word in the sequence. This is particularly crucial in transformer-based models, as they do not inherently capture sequential information. By incorporating positional data, the model can better understand the order and structure of elements in bibliographic references, such as distinguishing between the start and end of titles or authors' lists.

transformer_encoder

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{12}$$

$$\text{transformer\_out}(w_i) = \text{TransformerEncoderLayer}(\text{emb}_{\text{pos}}(w_i)) \tag{13}$$

$$\begin{aligned}
\text{X}_{\text{input}} &= \text{emb}_{\text{pos}}(w_i) && (14)\\
\text{X}_{\text{att}} &= \text{MultiheadAttention}(\text{X}_{\text{input}}, \text{X}_{\text{input}}, \text{X}_{\text{input}}) && (15)\\
\text{X}_{\text{dropout}} &= \text{Dropout}(\text{X}_{\text{att}}) + \text{X}_{\text{input}} && (16)\\
\text{X}_{\text{norm1}} &= \text{LayerNorm}(\text{X}_{\text{dropout}}) && (17)\\
\text{X}_{\text{intermediate}} &= \text{Linear1}(\text{X}_{\text{norm1}}) && (18)\\
\text{X}_{\text{output}} &= \text{Linear2}(\text{X}_{\text{intermediate}}) && (19)\\
\text{X}_{\text{dropout2}} &= \text{Dropout}(\text{X}_{\text{output}}) + \text{X}_{\text{norm1}} && (20)\\
\text{transformer_out}(w_i) &= \text{LayerNorm}(\text{X}_{\text{dropout2}}) && (21)
\end{aligned}$$

The transformer encoder layer employs multi-head attention mechanisms, enabling the model to focus on different parts of the sequence simultaneously. This multi-faceted approach is beneficial for bibliographic references, as it allows the model to capture complex relationships and dependencies between different parts of a reference, such as correlating authors with titles or publication years. The layer also includes several normalization and dropout steps, ensuring stable and efficient training.

Each of these models was carefully designed and optimized for reference segmentation, considering both accuracy in identifying reference components and computational efficiency.

4.4. Training

From the Giant corpus, described in Section 3, the training and validation sets were generated, using a Python script that transforms the XML-tagged reference into CONLL-BIO format. It is important to emphasize that each token representing punctuation will be marked with the PUNC class, as these elements are key to distinguishing the different components of a reference string. Below is an example of a reference tagged with a class.

Raw text string citation:

Ritchie, E. and Powell, Elmer Ellsworth (1907) Spinoza and Religion. The Philosophical Review, 16(3), p. 339. [online] Available from: http://dx.doi.org/10.2307/2177340.

Citation in CONLL-BIO format:

- Ritchie B-AUTHOR
- , B-PUNC
- E I-AUTHOR
- . B-PUNC
- and I-AUTHOR
- Powell I-AUTHOR
- , B-PUNC
- Elmer I-AUTHOR
- Ellsworth I-AUTHOR
- (B-PUNC
- 1907 B-YEAR
-) B-PUNC
- Spinoza B-TITLE
- and I-TITLE
- Religion I-TITLE
- . B-PUNC
- The B-CONTAINER-TITLE
- Philosophical I-CONTAINER-TITLE
- Review I-CONTAINER-TITLE
- , B-PUNC
- 16 B-VOLUME
- (B-PUNC
- 3 B-ISSUE
-) B-PUNC
- , B-PUNC
- p O
- . B-PUNC
- 339 B-PAGE
- . B-PUNC
- [B-PUNC
- online O
-] B-PUNC
- Available O
- from O
- : O
- http://dx.doi.org/10.2307/2177340 B-URL

This dataset comprises 125,801 records, representing all citation styles and document types. These were divided proportionally and randomly using the SciKit-Learn library (https://scikit-learn.org/, accessed on 28 December 2023) in Python, as follows:

1. 80% for training.
2. 10% for hyperparameter tuning.
3. 10% for performance evaluation.

In the process of developing and evaluating machine learning models, the partitioning of data into distinct sets for training, hyperparameter tuning, and performance evaluation plays a pivotal role in ensuring the model's effectiveness and generalizability. For this study, the dataset was divided into three subsets: 80% allocated for training, 10% for hyperparameter tuning, and the remaining 10% for performance evaluation. This distribution was chosen to provide a substantial amount of data for the model to learn from during the training phase, ensuring a deep and robust understanding of the task. The allocation of 10% for hyperparameter tuning allows for sufficient experimentation with model configurations

to find the optimal set of parameters that yield the best performance. Similarly, reserving 10% for the evaluation set ensures that the model's effectiveness is tested on unseen data, offering a reliable estimate of its real-world performance. This balanced approach facilitates a comprehensive model development cycle, from learning and tuning to a fair and unbiased evaluation, critical for achieving high accuracy and generalizability in bibliographic reference segmentation tasks.

The hyperparameters used for the three models (CRF, BiLSTM + CRF, Transformer + CRF) are shown in Table 2:

Table 2. Training parameters.

Parameter	Value
Learning Rate	0.003
Batch Size	1024
Maximum Epochs	150
Optimizer	AdamW
Patience	2

The selection of hyperparameters for the three models (CRF, BiLSTM + CRF, Transformer + CRF) was a meticulous process aimed at optimizing performance while efficiently utilizing available computational resources. As detailed in Table 2, critical parameters such as learning rate, batch size, maximum epochs, the optimizer used, and patience were carefully adjusted through experimental iteration until the models achieved the lowest possible loss per epoch. This iterative approach ensured that each model could learn effectively from the training data, adapting its parameters to minimize error rates progressively. The chosen learning rate of 0.003 and the batch size of 1024 were particularly instrumental in this process, striking a balance between rapid learning and the capacity to process a substantial amount of data in each training iteration. Additionally, the AdamW optimizer facilitated a more nuanced adjustment of the learning rate throughout the training process, further contributing to the models' ability to converge towards optimal solutions. The patience parameter, set at 2, allowed for early stopping to prevent overfitting, ensuring the models remained generalizable to new, unseen data. This strategic selection and tuning of hyperparameters reflect a deliberate effort to maximize the models' learning efficiency and performance, capitalizing on the computational resources to achieve the best possible outcomes in bibliographic reference segmentation tasks.

4.5. Model Evaluation

The evaluation of the Transformer + CRF, BiLSTM + CRF, and CRF models, was conducted using a subset of data specifically allocated for performance assessment, which constituted 10% of the dataset extracted from the Giant corpus, revealing differences in their performance. These differences manifest in general metrics and in specific class performance, providing a deep understanding of each model's effectiveness in bibliographic reference segmentation.

The selection of F-score, Precision, and Recall as evaluation metrics for our models, particularly in Named Entity Recognition (NER) tasks and bibliographic reference segmentation, follows established best practices within the field. These metrics are widely recognized in the state-of-the-art for their ability to provide a comprehensive assessment of a model's performance. In terms of these metrics, 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. This level of accuracy signifies an almost flawless capability of the model to correctly identify and classify the components of bibliographic references, underlining the effectiveness of the chosen metrics in capturing the nuanced performance of NER models in the specialized task of bibliographic reference segmentation.

On the other hand, both the Transformer + CRF and traditional CRF models showed equally but slightly lower results compared to BiLSTM + CRF, with F-scores of 0.9863 and

0.9854, respectively. These results suggest that, although effective, these models may not be as precise in capturing certain fine details in references as the BiLSTM + CRF, see Figure 4.

When examining the performance by class, interesting trends are observed. In categories like PUNC, URL, and ISSN, all three models demonstrated high effectiveness, with BiLSTM + CRF and Transformer + CRF even achieving perfect precision in several classes.

However, in categories like VOLUME and ISSUE, which may present greater challenges due to their lower frequency or greater variability in references, there is a noticeable decrease in the performance of Transformer + CRF and CRF, while BiLSTM + CRF maintains relatively high efficacy, see Table 3.

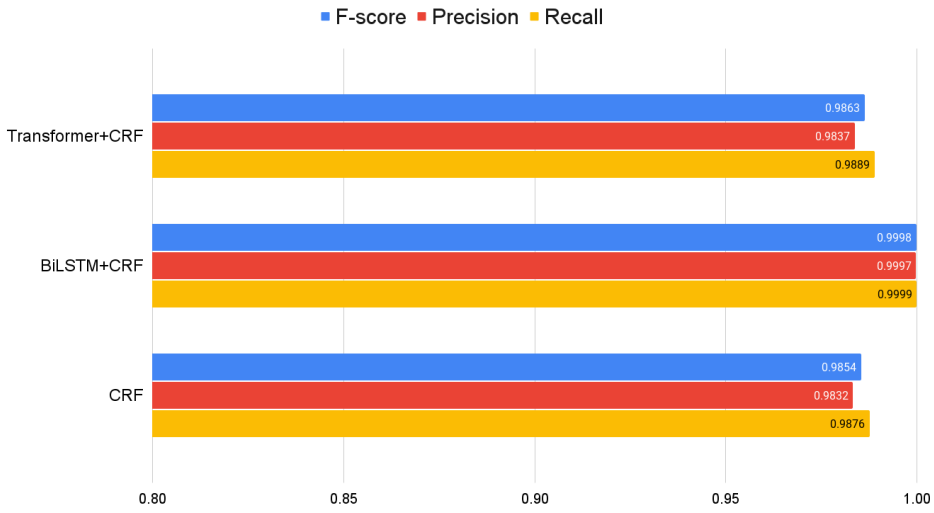

Figure 4. Overall performance metrics of the models.

Table 3. Comparative analysis of model performance across different classes.

Class	Transformer + CRF	BiLSTM + CRF	CRF
PUNC	1	**1**	1
AUTHOR	0.9799	**0.9997**	0.9771
TITLE	0.969	**0.9997**	0.9667
CONTAINER-TITLE	0.9512	**0.9997**	0.9484
YEAR	0.963	**0.9992**	0.9629
PUBLISHER	0.9837	**0.9998**	0.9834
DOI	0.9667	**1**	0.9642
URL	0.9987	**1**	0.9987
PAGE	0.9919	**1**	0.994
VOLUME	0.8049	**0.9936**	0.8064
ISSUE	0.8615	**0.9872**	0.8553
ISBN	0.9924	**1**	0.9922
ISSN	0.9714	**1**	0.9655

Values in bold represent the highest accuracy achieved for each class.

Notably, certain categories like VOLUME and ISSUE present a greater challenge for the models, with the BiLSTM + CRF showing a significant improvement over the other two models. This could reflect the contextual variability and complexity of these categories within bibliographic references.

5. Experiments and Results

An experiment was carried out on a test corpus that was totally different from the training corpus in order to assess the generalization and robustness of the produced models (CRF, BiLSTM + CRF, and Trasnformer + CRF). This corpus, known as CORA (described

in Section 3.1.2), consists of a wide range of bibliographic references and represents a significant challenge due to its diversity and differences from the training dataset.

The CORA corpus, with its distinctive feature of containing references with missing components, offers an ideal test scenario to evaluate the adaptability of the trained models to previously unseen data [7,15,16]. This unique characteristic of the corpus underscores the importance of model resilience in handling incomplete data, providing a rigorous test bed to assess the models' capability to adjust to new contexts and data structures. Such an evaluation is crucial for real-world applications, where bibliographic references often exhibit significant variability in format, style, and completeness.

In this section, we present the results obtained by applying the trained models to the CORA corpus. The same metrics used in the training evaluation—F-score, Precision, and Recall—were employed to maintain consistency and allow direct comparisons. Additionally, the performance by class for each model was analyzed, offering a view of their effectiveness in specific categories of bibliographic references.

The results obtained in this experimental scenario provide a thorough evaluation of each model's ability to generalize and adapt to new datasets. The following subsection details these results, offering a comparison of the performance of the models in the CORA corpus.

5.1. Results on the CORA Corpus

Evaluating the CRF, BiLSTM + CRF, and Transformer + CRF models on the CORA corpus provides insights into their ability to adapt and perform on a dataset different from the one used for their training. It is important to note that there are classes in CORA that are different or non-existent in the Giant dataset. Therefore, adjustments were made to align the CORA labels with those recognized by the models to ensure consistency in our evaluation. This alignment process is detailed in Section 3.2.2 and involves transforming certain CORA labels (e.g., 'PAGES' to 'PAGE') to match the class definitions used during model training. These adjustments are crucial for accurately assessing the models' performance across datasets and are summarized in Table 4, where each CORA entity is mapped to the corresponding entity recognized by the models.

Table 4. Adaptation to the CORA labels.

Cora Entity	Entity in Models
AUTHOR	AUTHOR
BOOKTITLE/JOURNAL	CONTAINER-TITLE
DATA	YEAR
PAGES	PAGE
PUBLISHER	PUBLISHER
VOLUME	VOLUME
TITLE	TITLE
TECH	<REMOVED>
INSTITUTE	<REMOVED>
EDITOR	<REMOVED>
NOTE	<REMOVED>

Regarding general metrics, the results show similar trends to those observed during the training phase. The BiLSTM + CRF model continues to demonstrate superior performance, with an F-score of 0.9612, a precision of 0.9653, and a recall of 0.9572, see Figure 5.

These results reaffirm the robustness of the BiLSTM + CRF in terms of accuracy and its ability to capture relevant contexts. Meanwhile, the Transformer + CRF and the CRF alone show slightly lower performance, with F-scores of 0.8957 and 0.8995, respectively.

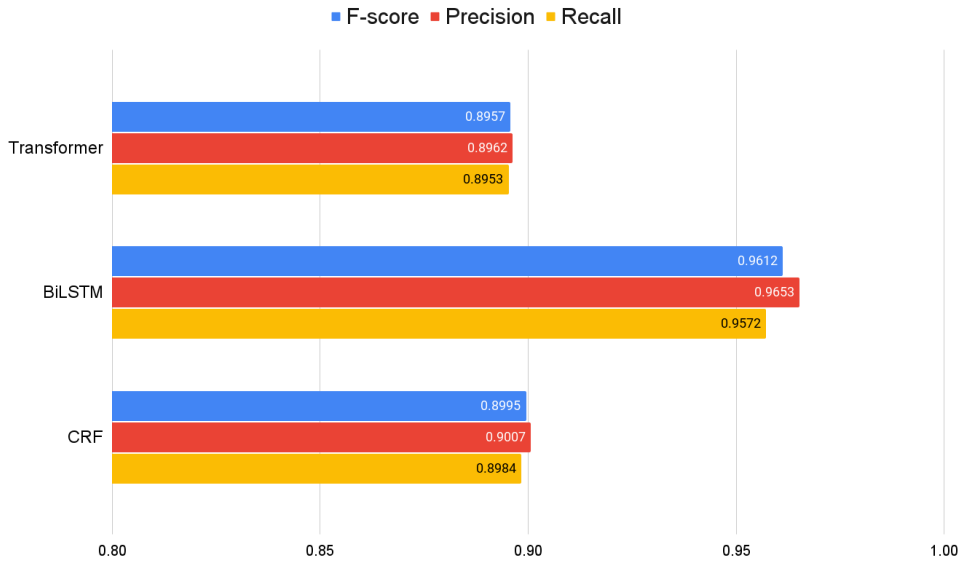

Figure 5. Overall performance metrics of the models - CORA Corpus.

These findings highlight the BiLSTM + CRF's superior adaptability and accuracy in dealing with a diverse set of bibliographic references, a critical aspect for real-world applications in digital libraries and information management systems.

The analysis by category reveals notable variations in performance among the models. While all variants achieve a perfect F-score in the PUNC category, there are significant differences in other categories. For instance, in the TITLE category, BiLSTM + CRF considerably outperforms its counterparts, with an F-score of 0.838 compared to 0.6096 (CRF) and 0.5441 (Transformer). Similarly, in categories like CONTAINER-TITLE, PUBLISHER, and VOLUME, BiLSTM + CRF shows a notably greater ability to correctly identify these elements, which could be attributed to its better handling of the variability and complexity of these categories, see Table 5.

Table 5. Comparative analysis of model performance across different classes - CORA.

Class	Transformer + CRF	BiLSTM + CRF	CRF
PUNC	1	**1**	1
AUTHOR	0.9517	**0.9876**	0.9325
TITLE	0.5441	**0.838**	0.6096
YEAR	0.982	**0.9885**	0.9804
CONTAINER-TITLE	0.1392	**0.619**	0.1937
PAGE	0.8652	**0.9483**	0.8803
VOLUME	0.3402	**0.8213**	0.3425
PUBLISHER	0.2629	**0.7931**	0.2903

Values in bold represent the highest accuracy achieved for each class.

On the other hand, it is interesting to note that in categories like YEAR and PAGE, all models show relatively high performance, indicating a certain uniformity in the structure of these categories that the models can effectively capture.

In summary, the results on the CORA corpus suggest that while the BiLSTM + CRF model is consistently superior in several categories, the differences in performance between the models become more pronounced in more complex and varied categories. This underscores the importance of choosing the right model based on the specific characteristics of the task and dataset. The superiority of the BiLSTM + CRF on CORA Corpus reinforces its potential as a reliable tool for bibliographic reference segmentation tasks, especially in environments where data diversity and complexity are high.

5.2. Considerations about the CORA Corpus

It is crucial to consider certain specific characteristics of the CORA corpus when evaluating the performance of the models. One of the most significant peculiarities is the presence of references with missing components, which represents a unique challenge for reference segmentation models. A representative example of this type of case is as follows (Listing 7):

Listing 7: Example of reference with missing components.

```
<author> M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M.
    Fahlman, O. Inganas and M.R. Andersson, </author> <
    container-title> J Appl. Phys., </container-title> <volume>
     76, </volume><pages>893, </pages> <year> (1994). </year>
```

In this example, the reference lacks the TITLE label, leading the model to erroneously inferring that the first tokens of CONTAINER-TITLE belong to TITLE. This situation affects the scores of both classes and highlights the challenge of handling incomplete references, which are common in the CORA corpus.

5.3. Discussion

The detailed performance analysis of Transformer + CRF, BiLSTM + CRF, and CRF models on the CORA corpus provides valuable insights into their efficacy in addressing the segmentation of bibliographic references across various classes. While the BiLSTM + CRF model exhibits a clear advantage in handling diverse and incomplete data sets, as evidenced by its superior performance across almost all categories, the results also shed light on the challenges and limitations faced by the other models.

The Transformer + CRF model, despite its innovative architecture designed to capture long-range dependencies and contextual nuances, struggles significantly with certain classes such as CONTAINER-TITLE and PUBLISHER, where it scores remarkably lower than its counterparts. This suggests a potential limitation in its ability to handle instances where contextual cues are sparse or irregular, common in real-world bibliographic data.

Conversely, the CRF model, while not achieving the high performance of the BiLSTM + CRF model, demonstrates a degree of resilience, outperforming the Transformer + CRF model in several classes. This indicates that traditional CRF models, despite their simpler architecture, can still be competitive, particularly in scenarios where the data structure benefits from their sequence modeling capabilities. However, its performance in critical categories such as TITLE and CONTAINER-TITLE remains suboptimal, highlighting the necessity for more sophisticated sequence modeling techniques to capture the complex patterns present in bibliographic references effectively.

The comparative analysis underscores the BiLSTM + CRF model's robustness and its capacity to adapt to the CORA dataset's irregularities, making it a potent tool for bibliographic reference segmentation tasks. Meanwhile, the observed performance disparities among the models underscore the imperative of selecting a modeling approach that is not only theoretically sound but also practically attuned to the specific challenges posed by the data. This entails a nuanced consideration of each model's architectural strengths and weaknesses, ensuring the chosen method aligns with the inherent complexities of bibliographic data encountered in digital libraries and databases.

The results demonstrate the application of these models in digital library management systems. Specifically, these models can automate manual validation processes for citation analysis, as mentioned in [26]. The integration of the BiLSTM + CRF model into existing systems enables libraries to segment and classify bibliographic references with an efficiency of 96.12% based on the F1 metric data from our experiments with the CORA corpus. This integration facilitates improvements in information retrieval and supports academic research activities.

Furthermore, these findings align with recent observations reported [27], which suggest that transformer models may not always excel in tasks such as Named Entity Recogni-

tion (NER) where understanding the specific structure and immediate context is crucial. This study reinforces the notion that while transformers offer advanced capabilities for capturing long-range dependencies, their performance in structured prediction tasks like bibliographic reference segmentation might not match that of models designed to navigate syntactic complexities more effectively, such as BiLSTM + CRF.

6. Conclusions

This study embarked on a comparative analysis of three models—Conditional Random Fields (CRF), Bidirectional Long Short-Term Memory with CRF (BiLSTM + CRF), and Transformer Encoder with CRF (Transformer + CRF)—for the task of bibliographic reference segmentation. Conducted with the Giant corpus for model training and the CORA corpus for evaluation, this research aimed to identify the most effective model for parsing the intricate structure of bibliographic references. The BiLSTM + CRF model demonstrated superior performance, particularly excelling in the precise delineation of bibliographic components such as TITLE, CONTAINER-TITLE, VOLUME, and PUBLISHER compared to the other models, as evidenced by its superior F-scores in these categories. The success of this model can be attributed to its effective management of the syntactic relationships within bibliographic data, which is essential given the structured nature of these references.

The findings of this research not only emphasize the efficacy of the BiLSTM + CRF model in addressing data imperfections but also underscore its potential utility in digital library systems and automated bibliographic processing tools. By ensuring high accuracy across diverse datasets, the BiLSTM + CRF model emerges as a robust solution for the management and processing of bibliographic information in academic libraries.

Furthermore, the findings from this study stress the importance of model selection tailored to the specific needs of the task and the dataset. The superior adaptability and accuracy of the BiLSTM + CRF model in managing diverse bibliographic formats and complex categories reinforce its suitability for real-world environments where data variability and complexity are high. These insights contribute significantly to our understanding of bibliographic reference segmentation and set the stage for future research to refine these models further, enhancing their practical applicability.

Looking to the future, the research pathway in this domain is rich with potential. An immediate direction involves delving into the bio-inspired mechanisms, particularly focusing on lateral inhibition mechanisms, given that the BiLSTM architecture, closely mirroring biological neural networks more than transformers, shows promise in reference segmentation tasks. The exploration of reference segmentation through lateral inhibition mechanisms [28] could provide a novel approach, building on the bio-inspired foundations established by the BiLSTM architecture. This could open avenues for enhancing the model's ability to manage the wide variety of bibliographic formats and styles more effectively. Assessing the model's performance across a broader spectrum of languages and bibliographic traditions is also essential for ensuring its global applicability and utility in digital library systems. These future endeavors promise not only to refine the capabilities of bibliographic reference segmentation models but also to broaden their practical applications, significantly contributing to advancements in digital library services and scholarly communication.

Author Contributions: Conceptualization, R.C.H. and J.-M.T.-M.; methodology, R.C.H. and R.P.E.; software, R.C.H.; validation, R.C.H.; formal analysis, R.C.H.; data curation, R.C.H.; investigation, R.C.H.; writing—original draft preparation, R.C.H., O.O.V.V. and A.M.S.; writing—review and editing, R.C.H., O.O.V.V. and J.-M.T.-M.; supervision, R.P.E., J.-M.T.-M., G.R.S. and O.O.V.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data sets used are open access and links have been provided in https://doi.org/10.7910/DVN/LXQXAO, accessed on 15 January 2024; https://ask.library.uic.edu/faq/345899, accessed on 17 January 2024; https://people.cs.umass.edu/~mccallum/data/cora-refs.tar.gz, accessed on 22 January 2024 [21]; https://flairnlp.github.io/, accessed on 29 December 2023. https://scikit-learn.org/, accessed on 28 December 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khabsa, M.; Giles, C.L. The number of scholarly documents on the public web. *PLoS ONE* **2014**, *9*, e93949. [CrossRef]
2. Ware, M.; Mabe, M. *The STM Report: An Overview of Scientific and Scholarly Journal Publishing*; International Association of Scientific, Technical, and Medical Publishers: The Hague, The Netherlands, 2015.
3. Bornmann, L.; Mutz, R. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *J. Assoc. Inf. Sci. Technol.* **2015**, *66*, 2215–2222. [CrossRef]
4. Becker, D.A.; Chiware, E.R. Citation Analysis of Master's Theses and Doctoral Dissertations: Balancing Library Collections With Students' Research Information Needs. *J. Acad. Librariansh.* **2015**, *41*, 613–620. [CrossRef]
5. Rizvi, S.T.R.; Dengel, A.; Ahmed, S. A Hybrid Approach and Unified Framework for Bibliographic Reference Extraction. *IEEE Access* **2020**, *8*, 217231–217245. [CrossRef]
6. Tkaczyk, D.; Collins, A.; Sheridan, P.; Beel, J. Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, Fort Worth, TX, USA, 3–7 June 2018; pp. 99–108. [CrossRef]
7. Grennan, M.; Beel, J. 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. In *Proceedings of the 8th International Workshop on Mining Scientific Publications*; Association for Computational Linguistics: Wuhan, China, 2020; pp. 27–35. Available online: https://aclanthology.org/2020.wosp-1.4/ (accessed on 10 January 2024).
8. Jain, V.; Baliyan, N.; Kumar, S. Machine Learning Approaches for Entity Extraction from Citation Strings. In *Proceedings of the International Conference on Information Technology*; Springer: Singapore, 2023; pp. 287–297.
9. Choi, W.; Yoon, H.M.; Hyun, M.H.; Lee, H.J.; Seol, J.W.; Lee, K.D.; Yoon, Y.J.; Kong, H. Building an annotated corpus for automatic metadata extraction from multilingual journal article references. *PLoS ONE* **2023**, *18*, e0280637. [CrossRef]
10. Bergmark, D. *Automatic Extraction of Reference Linking Information from Onlinedocuments*; Cornell University: Ithaca, NY, USA, 2000.
11. Hetzner, E. A simple method for citation metadata extraction using hidden markov models. In Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries, Pittsburgh, PA, USA, 16–20 June 2008; pp. 280–284.
12. Patro, S.; Wang, W. Learning Top- k Transformation Rules. In *Proceedings of the Database and Expert Systems Applications*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 172–186.
13. Peng, F.; McCallum, A. Information extraction from research papers using conditional random fields. *Inf. Process. Manag.* **2006**, *42*, 963–979. [CrossRef]
14. Lopez, P. GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 473–474.
15. Councill, I.G.; Giles, C.L.; Kan, M.Y. ParsCit: An open-source CRF Reference String and Logical Document Structure Parsing Package. In Proceedings of the 6th International Conference on Language Resources and Evaluation, Marrakech, Morocco, 26 May–1 June 2008; Volume 8, pp. 661–667. ECDL 2009
16. Prasad, A.; Kaur, M.; Kan, M.Y. Neural ParsCit: A deep learning-based reference string parser. *Int. J. Digit. Libr.* **2018**, *19*, 323–337. [CrossRef]
17. Rodrigues Alves, D.; Colavizza, G.; Kaplan, F. Deep Reference Mining From Scholarly Literature in the Arts and Humanities. *Front. Res. Metrics Anal.* **2018**, *3*. [CrossRef]
18. Tkaczyk, D.; Gupta, R.; Cinti, R.; Beel, J. ParsRec: A novel meta-learning approach to recommending bibliographic reference parsers. In Proceedings of the 26th AIAI Irish Conference on Artificial Intelligence and Cognitive Science Trinity College Dublin, Dublin, Ireland, 6–7 December 2018; Volume 2259, pp. 162–173. Available online: https://ceur-ws.org/Vol-2259/ (accessed on 3 February 2024).
19. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
20. Grennan, M.; Schibel, M.; Collins, A.; Beel, J. Giant: The 1-billion annotated synthetic bibliographic-reference-string dataset for deep citation parsing. In Proceedings of the 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science, Galway, Ireland, 5–6 December 2019; Volume 2563, pp. 260–271.
21. Anzaroot, S.; McCallum, A. A new dataset for fine-grained citation field extraction. In Proceedings of the ICML 2013 Workshop on Peer Reviewing and Publishing Models, Atlanta, GA, USA, 20 June 2013 .

22. Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; Vollgraf, R. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 54–59.
23. Lafferty, J.; McCallum, A.; Pereira, F.C. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*; Proceedings of the Eighteenth International Conference on Machine Learning; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2001.
24. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
26. Gutiérrez de la Torre, S.E.; Ortiz Reyes, J.V.; Escobar Farfán, J.I.; Bocanegra Esqueda, T.; Cid Carmona, V.; Escobar Vallarta, C.; Quiroa Herrera, M.L.; Romero Millán, C. Datos bibliométricos para las Ciencias Sociales y las Humanidades: Un método para el acopio, validación y análisis con herramientas de acceso gratuito. In Proceedings of the X Conferencia Internacional de Bibliotecas y Repositorios Digitales (BIREDIAL-ISTEC) (Modalidad Virtual, 25 al 29 de octubre de 2021), Virtual, 25–29 October 2021.
27. Yan, H.; Deng, B.; Li, X.; Qiu, X. TENER: Adapting transformer encoder for named entity recognition. *arXiv* **2019**, arXiv:1911.04474.
28. Mitrofan, M.; Păiș, V. Improving Romanian BioNER using a biologically inspired system. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 316–322.