Teanga Data Model for Linked Corpora

John P. McCrae, Priya Rani, Adrian Doyle, Bernardo Stearns
SFI Insight Centre for Data Analytics, Data Science Institute, University of Galway, Ireland
john@mccr.ae, priya.rani@insight-centre.org,
adrian.doyle@insight-centre.org, bernardo.stearns@insight-centre.org

Abstract

Corpus data is the main source of data for natural language processing applications, however no standard or model for corpus data has become predominant in the field. Linguistic linked data aims to provide methods by which data can be made findable, accessible, interoperable and reusable (FAIR). However, current attempts to create a linked data format for corpora have been unsuccessful due to the verbose and specialised formats that they use. In this work, we present the Teanga data model, which uses a layered annotation model to capture all NLP-relevant annotations. We present the YAML serializations of the model, which is concise and uses a widely deployed format, and we describe how this can be interpreted as RDF. Finally, we demonstrate three examples of the use of the Teanga data model for syntactic annotation, literary analysis and multilingual corpora.

Keywords: corpora, natural language processing, linked data, formats

1. Introduction

Corpus data is vital to modern natural language processing and is often annotated with many layers of extra information from part of speech to complex structural and semantic categories. There are several standards for publishing corpora including the Text Encoding Initiative (Ide, 1994, TEI) and the Linguistic Annotation Framework (Eckart, 2012, LAF), however, none of these have become widely accepted in natural language processing. In contrast, for lexico-semantic data, linked data models based on RDF have had great success through models such as OntoLex-lemon (McCrae et al., 2017; Cimiano et al., 2016). However, attempts to produce RDF models for representing corpus information such as the NLP Interchange Framework (Hellmann et al., 2013, NIF) and POWLA (Chiarcos, 2012) have had less success. A major reason for the failure of these models to have sufficient traction in NLP communities is that the RDF models adopted for linked data and the XML models used for TEI and LAF are very verbose and do not fit in with modern natural language processing pipelines. As such, most natural language processing data does not satisfy the FAIR principles, particularly in relation to reusability as the use of custom parsers, which may be difficult for others to reuse. Similarly, the adoption of a linked data paradigm will increase the findability and accessibility of the resource by providing methods where corpora can be connected with lexicographic, terminological and encyclopaedic resources.

In this paper, we introduce a new model called the Teanga[1][2] data model, which aims to provide a simple, low-overhead method for sharing text corpora and interacting with linked data. The Teanga data model can be simply serialized as JSON or YAML, allowing it to be easily loaded and worked with in modern programming languages. The Teanga data model also develops a new method of annotation called *layered annotation*, which combines the best of stand-off annotation and in-line (XML-style) annotation to enable data to be quickly handled. Finally, the Teanga data model defines a method of annotation that provides a conversion to RDF and can be converted into standard RDF. We note the Teanga JSON serialization is inspired by JSON-LD (Sporny et al., 2020), but is not directly a JSON-LD model. The Teanga data model is being developed as part of Teanga 2, a new platform for NLP based on the previous Teanga platform (Ziad et al., 2018).

The rest of this paper is as follows: firstly, we will introduce the Teanga data model and layered annotation and then we will describe the technical implementation of the model, including serialization as YAML, an implementation in Python and the conversion to RDF. We will then provide three examples of conversions of data from Universal Dependencies (de Marneffe et al., 2021), conversion of TEI data, such as from the ELTeC (Schöch et al., 2021) corpus, and an example of parallel corpora with word-level alignment data. We will then conclude with a discussion of the Teanga data model in comparison to other corpus models.

[1] Teanga is Irish for tongue/language and is pronounced tʲanga

[2] https://teanga.io/

LDL: Linked Data in Linguistics @ LREC-COLING 2024, pages 66–74
20-25 May, 2024.

2. Design of the model

2.1. Layered Annotation Model

A corpus in the Teanga data model, as depicted in Figure 1, is composed of a metadata section and a list of documents. Each of these documents has layers that are defined in the metadata layers and may have some or all of these layers. All documents must have at least one *character* layer, which consists of a single Unicode string containing the text of the document. This means that Teanga preserves the plain text version of the document, in contrast to XML annotation where annotations must be inserted into the document. Currently, Teanga only supports text corpora, but the introduction of new base layer types would allow the model to extend to multimodal corpora. The remaining layers of annotation consist of a reference mechanism and (optionally) a data value. The referencing mechanism refers to an annotation in another layer (the *base layer*), which is defined in the metadata. For character layers, the elements are the Unicode characters in the layers. All indexes in layers start from zero. The referencing mechanisms are as follows:

- **Span Layer**: A span layer gives two indexes corresponding to the start and end of the annotation.
- **Division Layer**: The division layer divides the base layer into non-overlapping segments
- **Element Layer**: An element layer refers to a single element in the base layer
- **Sequence Layer**: A sequence layer corresponds to the annotation layer in a one-to-one manner so that there is one annotation for each element of the base layer.

In most cases, a span layer is used to divide the lower layer into words and other annotations are based on this word layer. Division layers are used to divide the text by sentences, paragraphs or chapters.

Each annotation in a layer must have the same data value, the values are defined as follows:

- **None**: No data is associated with an annotation, for example, in tokenization.
- **String**: A single string is associated with each annotation
- **Enumeration**: The annotation may have one value from a list of values given in the metadata section
- **Link**: A reference is defined to another annotation in the same layer, or in a secondary layer called the *target layer*.
- **Typed Link**: Combines the data of the enumeration and the link layer.

In addition to layers, each document or layer may have any number of *meta-properties*. The most important of which is the `_uri` property which gives the URI to interpret the document as linked data.

Each document in a Teanga corpus is associated with an identifier that ensures that the document content is valid. This check means that if the text content is changed, we can detect this a not proceed with annotations that have become invalid. The methodology for deducing the identifier is as follows: each document is indexed by initial characters the Base64 encoding of the SHA-256 of the UTF-8 representation of the text. The text representation consists of all character layers ordered alphabetically by their key with the key appended before the text. Keys and text should be separated by a zero byte (Unicode 0000). In most cases, the key should be at least four characters long and should be the shortest representation that is unique in the corpus. As such, it is not possible to have documents with duplicate text in a Teanga corpus. This can be avoided if necessary by adding an extra field with an identifier.

Finally, each corpus is associated with an *order* that gives the order of the documents in the corpus. This order is simply the list of identifiers in the document. This may be omitted in some serializations if the order is implicit in the serialization, however, if given it overrides the order in the serialized document.

3. Technical Implementation

The preferred method for representing Teanga corpora is as YAML documents. Teanga documents can also be easily represented as JSON documents. We provide a Python implementation of the Teanga model that ensures that models are easily serialized and provides useful features. In addition, a database model is provided for serializing and sharing the models as compact binary models. Further, we support the export of the model into RDF in both a generic method and methods compatible with NIF (Hellmann et al., 2013) and Web Annotation (Sanderson et al., 2017).

3.1. YAML form

The preferred serialzation of Teanga is as YAML. Teanga YAML documents consist of a dictionary with one special key `_meta`, an optional second special key `_order` and the remaining keys consist of the document identifier and a dictionary of the layers in the document. The `_order` key is normally omitted in YAML as the order of documents

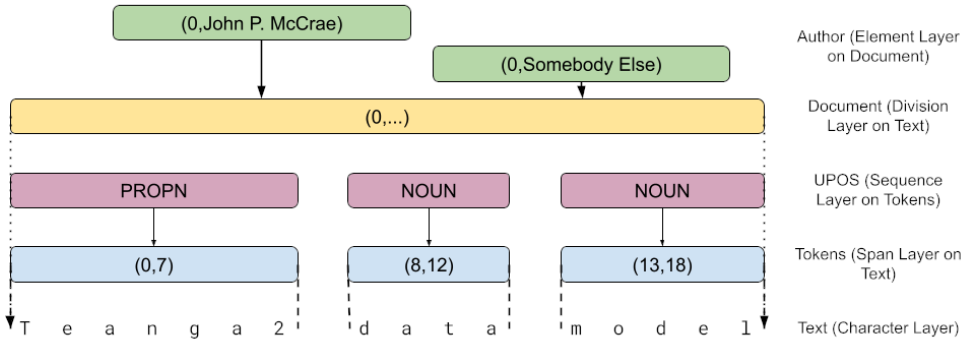

Figure 1: An example of the annotation layers of a Teanga document

in the text can be inferred by the order in the document. For example, a simple Teanga YAML document is as follows:

```
_meta:
    text:
        type: characters
Lz1r:
  text: This is an example document.
```

Each field of the metadata may have the following values

- `type`: The type of the layer (referencing mechanism)
- `base`: The name of the base layer (omitted for character layers)
- `data`: The data type. A list of values indicates an enumeration
- `target`: The target layer
- `link_types`: The enumeration of values used for links
- `default`: A default value for this layer if omitted
- `_uri`: The URI of the RDF property that documents this layer

The `_order` element of a corpus is a simple list of document IDs. It is not required in YAML and may instead be inferred from the order of the words in the document

Each non-character layer consists of a list of lists[3], where each list consists of zero, one or two indexes and the data consisting of an optional integer for the target of a link and a string for the data or enumeration type. As such, each element may consist of one to five elements[4]. It is important to note that the indexes refer to the order of annotations in the base layer, not the absolute character index, unless the base layer is a character layer. For division layers, the index indicates the start index of each section. So for example to provide tokens and sentences we may have the following document.

```
_meta:
  text:
    type: characters
  tokens:
    type: span
    base: text
  sentences:
    type: div
    base: tokens
hDRz:
  text: Hello there! Goodbye!
  tokens: [[0,5], [6,11], [11,12],
   ↪  [13,20], [20,21]]
  sentences: [0, 3]
```

It is important to note here that the indexes in the sentence layer are in terms of the tokens, so the second sentence starts from the 4th token (index=3) and this can be mapped into characters by reference to the token layer.

3.2. Python Implementation

Teanga is provided as a Python library on GitHub[5] this library supports the basic operation of the library including adding and removing documents and updating metadata layers. In addition, it provides support for mapping indexes from a base layer to lower layers, which is a specific challenge as complex multi-layer annotations may make it difficult to reach the actual characters the annotations are referring to.

[3] This efficient representation cannot be supported by JSON-LD

[4] Zero elements are allowed but meaningless

[5] https://github.com/teangaNLP/teanga2

In addition to providing a strong in-memory version of the Teanga data model, a secondary implementation[6] in Rust using the Sled[7] library provides a simple method for working with large corpora in Teanga. This persists large corpora to disk, allowing them to be searched and queried efficiently. A full interface is available for this in Python and as such there is minimal change to use this version of the interface. Due to this technology, it will be possible for Teanga to handle very large corpora, of the order of billions of tokens, and load and parse such corpora rapidly. We will further investigate this alongside tools for improving query time of the corpora based on use cases of the systems in future work focussed on these aspects.

3.3. Conversion to RDF

Support for conversion to linked data is a key goal of Teanga and it is expected that Teanga corpora could be used as targets for linking of other resources. If `_uri` properties are given for layers these can be used to map the resource to RDF. Each document in the model is given a URI based on its identifiers and these are included in the fragment identifier. So, for example, we can indicate an identifier as follows for a document available at `http://www.example.com/corpus.yaml`.

```
_meta:
  text:
    type: characters
    _uri: https://teanga.github.io/\
teangaNLP/teanga.rdf#text
hDRz:
  text: Hello there! Goodbye!
```

Is converted to Turtle as follows:

```
<http://www.example.com/corpus.yaml#hDRz>
  teanga:text "Hello there! Goodbye!"
```

Annotations in Teanga are modelled with the use of two special properties `teanga:idx`, `teanga:ref` and `teanga:data`[8], which give the order of the annotation and a reference to the base layer and the data, respectively.

Following RFC 5147, references to text layers can be made with `char=` elements in the fragment, for example:

```
<#hRDz>
  ex:tokens [
    teanga:idx 0 ;
    teanga:ref <#hRDz&char=0,5>
  ] .
```

References to any other layer can be made with `n=` fragment.

The default URI for a document is given by adding the Teanga document identifier to the URI, but can alternatively be specified by giving a `_uri` property on the individual document.

In addition to this direct export to RDF using the Teanga RDF vocabulary, it is also possible to export to NIF and WebAnnotation style vocabularies. The RDF generated in these exports is generally more verbose than the Teanga RDF model. For example, the NIF export looks like this:

```
<#hRDz&char=0,5> a
 ↪  nif:OffsetbasedString ;
  nif:anchorOf "Hello" ;
  nif:beginIndex 0 ;
  nif:endIndex 5 ;
  rdf:value ex:tokens .
```

Similarly, the annotation in the WebAnnotation model is as follows in JSON-LD:

```
{
  "@context":
"http://www.w3.org/ns/anno.jsonld",
  "id": "#anno_1",
  "type": "Annotation",
  "body": {
    "value": {
      "@id":
"http://www.example.com/tokens"
    }
  },
  "target": {
    "source": "#hRDz",
    "selector": {
      "type":
       ↪  "TextPositionSelector",
      "start": 0,
      "end": 5
    }
  }
}
```

Note that we use the fully expanded URI for `ex:tokens` in this example.

4. Examples

We present three examples of NLP data and how they can be represented in Teanga by means of examples. Conversion tools for these formats are already published or under development.

4.1. CoNLLU Data

CoNLLU data format is the representation of the linguistic data developed to train the dependency parser once at a time for many different languages.

[6] https://github.com/teangaNLP/teanga.rs

[7] https://docs.rs/sled/latest/sled/

[8] The namespace `teanga` is defined as `https://teanganlp.github.io/teanga2/teanga.rdf`

The annotation of the CoNLLU is encoded in plain text format where a break line or LF character is used for representing a new line. The data has three different types of lines: - comment line: This line represents any sentence-level comments. It is represented with '#' and is usually at the beginning of the sentence. - token/words: This line contains the annotation of a word/token/node in 10 fields separated by single tab characters - newline: This is a blank line at the end of each sentence, which indicates the sentence boundary.

In the Teanga model, we include the UD data with the same annotation features; however, the annotation representation starts with the character level. As described in Section 3.1, we convert the CoNLLU data in Teanga model format, which is represented in YAML format as shown in the following example.

This is converted to Teanga as follows. We note that the header information is fixed and as such the document has a similar size without the header[9]

```
_meta:
  text:
    type: characters
  tokens:
    base: text
    type: span
  comm:
    base: text
    type: characters
    data: string
  upos:
    data: ["ADJ", "ADP", "ADV",
     ↪  "AUX", "CCONJ", "DET",
     ↪  "INTJ", "NOUN", "NUM",
     ↪  "PART", "PRON", "PROPN",
     ↪  "PUNCT", "SCONJ", "VERB",
     ↪  "X" ]
    base: tokens
    type: seq
sjKY:
  text: _Bhojpuri text_
  tokens: [[0, 7], [8, 9], [10, 20],
   ↪  [21, 24], [25, 26]]
  comm: lokaraṁjana ā sāṁskrtika
   ↪  gīta -
  upos: ["NOUN", "CCONJ", "ADJ",
   ↪  "NOUN", "PUNCT" ]
u40k:
  text:  _Bhojpuri text_
  tokens: [[0, 3], [4, 5], [6, 13],
   ↪  [14, 17], [18, 19]]
  comm: āīṁ ā saparivāra āīṁ .
  upos: ["VERB", "CCONJ", "NOUN",
   ↪  "VERB", "PUNCT"]
```

In the above example [10], we can see that each sentence of the text file is tokenised at the character level and consists of only text. The rest of the features, including the ten fields, are categorised in the span layers as shown in `upos`[11]; similarly, the other morphological features will be included in the span layer. As we tokenize the text into character, there is no need to include the newline to show the sentence boundary, as in CoNLLU data. The tanga format also makes it easier to extract each feature of the text through the span layers. For example, if a task needs to use only parts of speech information of the given text, then the user can easily extract only the upon layer of the text rather than the whole document.

4.2. TEI Conversion

TEI tags can be used to annotate a variety of text features, as well as information about a text:

```
<div type="prose">
 <p>
  <supplied>B</supplied>ui
   ↪  oeng<expan>us</expan>
   ↪  hindaidqi naile inachotlud
   ↪  confacca ni hinningin chuici
   ↪  <expan>ar</expan> crannsiuil
   ↪  do.
  </p>
</div>
```

When a TEI-encoded text is converted to Teanga, the character layer consists of the text with all XML tags removed leaving only the text of the document as a single string of characters. The information which had been encoded using these TEI tags is instead preserved in a span layer according to the Teanga data model. Thus, the information represented in the TEI encoded text above may be represented in Tenga as follows:

```
_meta:
  text:
    type: characters
  div:
    type: span
    base: text
  div_type:
    type: element
```

[9] Note the original example uses Devanagari script but these could not be reproduced in the PDF and have been replaced with 'Bhojppuri text'. This is not a limitation of Teanga.

[10] The example is taken from the Bhojpuri UD data https://github.com/UniversalDependencies/UD_Bhojpuri-BHTB/tree/master".

[11] Note that it is possible to assign URIs to each of these values and as such, they can be mapped to other schemes such as OLiA or LexInfo

```
    base: div
    data: string
  p:
    type: span
    base: text
  supplied:
    type: span
    base: text
  expan:
    type: span
    base: text
7nkN:
  text: Bui oengus hindaidqi naile
   ↪  inachotlud confacca ni
   ↪  hinningin chuici ar crannsiuil
   ↪  do.
  p: [[0, 84]]
  div: [[0, 84]]
  div_type: [[0, "prose"]]
  supplied: [[0,1]]
  expan: [[8,10], [67,69]]
```

Aside from formatting tags like `<div>` or `<p>`, TEI annotation can also be used to preserve specific information about small portions of a text, often at word-level or smaller granularity. Repositories containing historical texts, for example, may use TEI tags to identify snippets of digital text which were added or changed by modern editors, but which were not present in an earlier manuscript. The example above, which was taken from *Thesaurus Linguae Hibernicae* (Kelly et al., 2006), uses `<supplied>` tags to identify text which has been supplied by the editors, and `<expan>` tags to show where manuscript abbreviations have been expanded. As with word-level annotations, this kind of information is captured by the Teanga data model in the span layer, though such annotations often apply to portions of text at a sub-word level. We also see how attributes of XML elements are mapped to layers that are dependent on the tag annotation, for example, the `div_type` layer represents the `type` attribute of the `div` tag.

```
<title>The Sign of Four</title>
<author>Doyle, Arthur Conan
 ↪  (1859-1903).</author>
<publisher>London: Spencer
 ↪  Blackett</publisher>
<date>1890</date>
<ref target=
"http://archive.org/detail..."/>
```

Where TEI tags are used to annotate metadata which is unrelated to any specific span of text, for example, information pertaining to authorship or publishing (as shown above), this can be preserved in the Teanga data model at the document level. This is done by creating a `document` layer which refers to all information in the text.

```
_meta:
  text:
    type: characters
  document:
    type: div
    base: characters
    default: [[0]]
  title:
    type: seq
    base: document
abcd:
  text: "..."
  title: ["The Sign of Four"]
  author: ["Doyle, Arthur Conan
   ↪  (1859-1903)."]
```

Note that by specifying the `default` of the `document` layer as `[[0]]`, we give a default value of a division that starts at character 0 and ends at the end of the document. More complex linguistic annotations, such as those found in Level 2 of ELTeC (Schöch et al., 2021), can be modelled in a similar manner to what is done in the UD example in section 4.1.

4.3. Parallel Texts

Word alignment is the task of assigning words from one sentence (the source sentence) to words in a target sentence when given two parallel sentences that are translations of each other. Typically, the datasets for this task are distributed as bitext corpus, for example, a few sentences extracted from the Spanish-English Europarl v7 corpus:

```
¿Hay alguna objeción ? ||| Are there
 ↪  any comments ?
Muchas gracias ||| Thank you very
 ↪  much .
Apruebo esta petición. ||| I agree
 ↪  with this request.
```

When converting parallel sentences into Teanga, each sentence is represented through a character layer, with one layer for the source language and another for the target language. Following this, since tokenization is required for the alignment task, each character layer is annotated with a token span layer. Ultimately, the word alignments are annotated as an element-linking layer, connecting an element in the source tokens layer to a corresponding element in the target tokens layer.

```
_meta:
    align:
        type: element
        base: en_tokens
        data: link
        target: de_tokens
```

```
    en:
        type: characters
    en_tokens:
        type: span
        base: en
    es:
        type: characters
    es_tokens:
        type: span
        base: es

8I9N:
    es: ¿Hay alguna objeción?
    en: Are there any comments?
    es_tokens: [[0, 4], [5, 11],
     ↪  [12, 20], [20, 21]]
    en_tokens: [[0, 3], [4, 9], [10,
     ↪  13], [14, 22], [22, 23]]
    align: [[0, 0], [1, 2], [2, 3],
     ↪  [3, 4]]
JmZn:
    es: Muchas gracias.
    en: Thank you very much.
    es_tokens: [[0, 6], [7, 14],
     ↪  [14, 15]]
    en_tokens: [[0, 5], [6, 9], [10,
     ↪  14], [15, 19], [19, 20]]
    align: [[0, 2], [0, 3], [1,0],
     ↪  [1,1], [2,4]]
SQ/9:
    es: Apruebo esta petición.
    en: I agree with this request.
    es_tokens: [[0, 7], [8, 12],
     ↪  [13, 21], [21, 22]]
    en_tokens: [[0, 1], [2, 7], [8,
     ↪  12], [13, 17], [18, 25],
     ↪  [25, 26]]
    align: [[0, 0], [0, 1], [0, 2],
     ↪  [1, 3], [2, 4], [3, 5]]
```

In this example, we highlight the capability of Teanga to link elements in different layers of annotation, facilitating the representation of linked data in a coherent and interconnected manner, demonstrating its broader potential in handling complex annotation relationships within parallel texts. Further, we note that sentence alignment can also be modelled the same way if sentence annotations are available as in Section 3.1

5. Related Work and Discussion

Teanga is a new model for annotated corpora that aims to be able to represent all kinds of natural language processing data in a single, consistent manner. The most widely used formats for annotated corpora are either limited models, such as CoNLL, which can only represent token-level annotations and links between tokens (dependency parses) and as demonstrated Teanga can represent these kinds of data in a manner that is not substantially more verbose than these specific formats. As such, Teanga is a flexible data model that can be parsed without the need for external libraries except for a YAML parser which is widely available (although the Teanga library provides some additional features). Thus, this avoids the development of custom extensions of formats such as CoNLL (Chiarcos and Glaser, 2020; Graën et al., 2019), which requires the development of new parsers and avoids the risk of using proprietary formats that may be hard to access in the future.

The most widely used model that allows for general annotation of a corpus is TEI (Ide, 1994), however, this is a model based on XML and as such is destructive of the original text content. Further, extensions on TEI are not easy to write and the interface with RDF and linked data is not clear (Burrows et al., 2021). Also, as demonstrated Teanga is able to efficiently and correctly represent complex annotations found in TEI.

The Teanga data model is more closely related to attempts to create linked data corpus models. Two of these models have risen to particular prominences. Firstly, the NLP Interchange Format introduced by Hellmann et al. (2013) has seen adoption for tasks such as named entity recognition (Röder et al., 2014), question answering (Latifi and Sànchez-Marrè, 2013) and frame semantics (Alexiev and Casamayor, 2016). However, this model proves very verbose in practical applications and the project is not actively maintained anymore, with version 2.0 of the model being released in 2013 and very few updates in any of the provided tooling since 2016.

The Web Annotation data model (Sanderson et al., 2017), was introduced as a model for annotating documents on the web using RDF. Web annotations consist of annotations that link bodies with targets. The body can be either a literal value or structured content and is used to give the value of the annotation. The targets can be selected by various methods including character offsets, as well as through mechanisms such as XPointer (for XML documents). This annotation is used by the INCEpTION (Klie et al., 2018) platform for annotating documents. Teanga annotations are exportable to Web Annotation, however, the format is generally much more verbose than the Teanga model.

6. Conclusion

In this paper, we have presented a data model for Teanga, a new framework for NLP based on the previous Teanga model (Ziad et al., 2018). The

layer annotation model proposed by this model allows the representation of all NLP-relevant corpus data and does so in a manner that is efficient and readable. Further, this framework integrates with linked data, both as a linked data format in its own right and also by exporting to other RDF serializations such as Turtle and JSON-LD. This data model will simplify the publishing corpora as linked data, by providing tooling and a self-documenting format that satisfies FAIR principles.

7. Acknowledgements

This work has been supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics.

8. Bibliographical References

Vladimir Alexiev and Gerard Casamayor. 2016. FN goes NIF: integrating FrameNet in the NLP interchange format. In *Proceedings of the LDL 5th Workshop on Linked Data in Linguistics: Managing, Building and Using Linked Language Resources*, pages 1–10.

Toby Burrows, Matthew Holford, David Lewis, Andrew Morrison, Kevin Page, and Athanasios Velios. 2021. Transforming TEI manuscript descriptions into RDF graphs. *Graph Data-Models and Semantic Web Technologies in Scholarly Digital Editing*, 15:143.

Christian Chiarcos. 2012. POWLA: modeling linguistic corpora in OWL/DL. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, volume 7295 of *Lecture Notes in Computer Science*, pages 225–239. Springer.

Christian Chiarcos and Luis Glaser. 2020. A tree extension for CoNLL-RDF. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7161–7169, Marseille, France. European Language Resources Association.

Philipp Cimiano, John P. McCrae, and Paul Buitelaar. 2016. Lexicon model for ontologies: Community report. Technical report.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Comput. Linguistics*, 47(2):255–308.

Kerstin Eckart. 2012. A standardized general framework for encoding and exchange of corpus annotations: The linguistic annotation framework, LAF. In *11th Conference on Natural Language Processing, KONVENS 2012, Empirical Methods in Natural Language Processing, Vienna, Austria, September 19-21, 2012*, volume 5 of *Scientific series of the ÖGAI*, pages 506–515. ÖGAI, Wien, Österreich.

Johannes Graën, Tannon Kew, Anastassia Shaitarova, and Martin Volk. 2019. Modelling large parallel corpora: The zurich parallel corpus collection. In *Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. 2013. Integrating NLP using linked data. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, volume 8219 of *Lecture Notes in Computer Science*, pages 98–113. Springer.

Nancy Ide. 1994. Encoding standards for large text resources: The text encoding initiative. In *15th International Conference on Computational Linguistics, COLING 1994, Kyoto, Japan, August 5-9, 1994*, pages 574–578.

Patricia Kelly, Niall Brady, and Hugh Fogarty. 2006. TLH: Thesaurus Linguae Hibernicae. Online Resource.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Majid Latifi and Miquel Sànchez-Marrè. 2013. The use of NLP interchange format for question answering in organizations. In *Artificial Intelligence Research and Development - Proceedings of the 16th International Conference of the Catalan Association for Artificial Intelligence, Vic, Catalonia, Spain, October 23-25, 2013*, volume 256 of *Frontiers in Artificial Intelligence and Applications*, pages 235–244. IOS Press.

John P. McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. 2017. The ontolex-lemon model: development and applications. In *Proceedings of eLex 2017*, pages 587–597.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. 2014. N^3 - A collection of datasets for named entity recognition and disambiguation in the NLP interchange format. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 3529–3533. European Language Resources Association (ELRA).

Robert Sanderson, Paolo Ciccarese, and Benjamin Young. 2017. Web Annotation Data Model. W3C Recommendation. W3C Recommendation.

Christof Schöch, Roxana Patras, Tomaž Erjavec, and Diana Santos. 2021. Creating the european literary text collection (eltec): Challenges and perspectives. *Modern Languages Open*.

Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, Pierre-Antoine Champin, and Niklas Lindström. 2020. JSON-LD 1.1: A JSON-based Serialization for Linked Data. W3C Recommendation. W3C Recommendation.

Housam Ziad, John P. McCrae, and Paul Buitelaar. 2018. Teanga: A linked data based platform for natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).