3 The Core Specification

The DMLex Core is for monolingual lexical resources, where headwords, definitions, examples etc. are all in one and the same language.

3.1 Data types

3.1.1 `LexicographicResource`

A data set which can be viewed and used by humans as a dictionary and - simultaneously - ingested, processed and understood by software agents as a machine-readable database.

> **Warning**
>
> The correct name of this data type in DMLex is *lexicographic* resource, not *lexical*.

Properties:

- `language` (optional, IETF language code)
  - The language of headwords, definitions, examples.

    > **Note**
    >
    > DMLex is based on the assumption that all headwords in a single lexicographic resource are in the same language, and that definitions and examples, if any occur in the resource, are in that language too. The `language` property informs potential users of the resource which language that is.

- `transcriptionScheme` (optional, IETF language code, eg. `en-fonipa` for English IPA)
  - The scheme (e.g. IPA) in which the `transcription` property of `Pronunciation` objects is given.

    > **Note**
    >
    > DMLex is based on the assumption that, if you do use any pronunciation transcriptions in your (monolingual) lexicographic resource, they all follow the same scheme. The `transcriptionScheme` tells potential users of the lexicographic resource which scheme that is.

- `metadataLanguage` (IETF language code)
  - The language of the display captions of metadata items.

Children:

- `Entry` (one or more)
- `PartOfSpeechValue` (zero or more)
- `LabelValue` (zero or more)
- `InflectedFormCaption` (zero or more)
- `Mapping` (zero or more)

Example 1. Example of a lexicographic resource

```
LexicographicResource: language="en"
    Entry: headword="aardvark"
        Sense: ...
        Sense: ...
    Entry: headword="abacus"
        Sense: ...
        Sense: ...
    Entry: headword="abandon"
        Sense: ...
        Sense: ...
```

3.1.2 `Entry`

A part of a lexicographic resource which contains information related to exactly one headword.

Child of:

- `LexicographicResource`

Properties:

- `headword` (non-empty string)
  - The headword can be a single word, a multi-word expression, or any expression in the source language which is being described by the entry in the lexicographic resource.
- `homographNumber` (number, optional)

> **Note**
>
> Entries do not have an explicit listing order. An application can imply a listing order from a combination of the headword and the homograph number.

Children:

- `PartOfSpeech` (zero or more)
- `Label` (zero or more)
- `Pronunciation` (zero or more)
- `InflectedForm` (zero or more)
- `Sense` (zero or more)

Example 2. Example of a lexicographic entry

```
Entry: headword="bank", homographNumber=1
    PartOfSpeech: ...
    Pronunciation: ...
    Sense: ...
    Sense: ...
```

3.1.3 `Sense`

A part of an entry which groups together information relating to one of the (possibly multiple) meanings (or meaning potentials) of the entry's headword.

> **Note**
>
> An **entry** is a container for formal properties of the headword such as orthography, morphology, syntax and pronunciation. A **sense** is a container for statements about the headword's semantics.

Child of:

- `Entry`

Properties:

- `listingOrder`
    - Can be implicit from the serialization.
- `indicator` (optional, non-empty string)
    - A short statement that indicates the meaning of a sense and permits its differentiation from other senses in the entry.

    > **Note**
    >
    > Indicators are sometimes used in dictionaries as "mini-definitions" instead of or in addition to regular definitions. They are short, one-word or two-word paraphrases of the sense. Their purpose is to allow the (human) user to find the desired sense quickly.
- `definition` (optional, non-empty string)
    - A long statement that describes and/or explains the meaning of a sense.

    > **Note**
    >
    > In DMLex, the term *definition* encompasses not only formal sciency definitions, but also less formal explanations.

Children:

- `Label` (zero or more)
- `Example` (zero or more)

Example 3. Example of an entry with two senses, each having both an indicator and a definition

```
Entry: headword="bank"
    Sense: indicator="financial institution", definition = "an institution which..."
        Example: ...
    Sense: indicator="riverside", definition = "an earthen mound which..."
        Example: ...
```

3.1.4 `PartOfSpeech`

Any of the word classes to which a lexical item may be assigned, e.g. noun, verb, adjective, etc.

Child of:

- `Entry`

Properties:

- `value` (non-empty string)
    - Must be one of the values defined by instances of `PartOfSpeechValue`.
- `listingOrder`
    - Can be implicit from the serialization.

> **Note**
>
> If you want to model other grammatical properties of the headword besides part of speech, such as gender (of nouns) or aspect (of verbs), the recommended way to do that in DMLex is to conflate them to the part-of-speech label, for example "masculine noun" and

"feminine noun", or "perfective verb" and "imperfective verb".

Example 4. Example of a part of speech

```
LexicographicResource
  PartOfSpeechValue: value="n", displayValue="noun"
  Entry: headword="aardvark"
    PartOfSpeech: value="n"
```

3.1.5 `Label`

An indication of some restriction on the use of the lexical item such time (old-fashioned, neologism), region (dialect), register (formal, colloquial), domain (medicine, politics) or grammar (singular-only).

Child of:

- `Entry`
- `Sense`
- `Pronunciation`
- `InflectedForm`

Properties:

- `value` (non-empty string)
  - Must be one of the values defined by instances of `LabelValue`.
- `listingOrder`
  - Can be implicit from the serialization.

Example 5. Example of a label

```
LexicographicResource
  LabelValue: value="sl", displayValue="slang"
  Entry: headword="bitch"
    Sense: definition="a female dog"
    Sense: definition="a woman who is difficult to get along with"
      Label: value="sl"
```

3.1.6 `Pronunciation`

Information about the pronunciation of its parent.

Child of:

- `Entry`
- `InflectedForm`

Properties:

- At least one of:
  - `transcription` (non-empty string)

    Note

    The scheme of the transcription (e.g. IPA) can be communicated to users of the lexicographic resource through the `transcriptionScheme` property of the `LexicographicResource` object.
  - `recording` (string: name or URL of a sound file)
- `listingOrder`
  - Can be implicit from the serialization.

Children:

- `Label` (zero or more)

Example 6. Example of pronunciation given as transcription

```
Entry: headword="aardvark"
    Pronunciation: transcription="a:rdva:rk"
```

Example 7. Example of pronunciation given as a sound file

```
Entry: headword="aardvark"
    Pronunciation: recording="aardvark.mp3"
```

Example 8. Example of pronunciation given both ways

```
Entry: headword="aardvark"
    Pronunciation: transcription="a:rdva:rk", recording="aardvark.mp3"
```

3.1.7 `InflectedForm`

An inflected headword is a form of the inflectional paradigm of its parent.

Child of:

- `Entry`

Properties:

- `caption` (non-empty string) e.g. 'plural'
    - Must be one of the values defined by instances of `InlectedFormCaption`.
- `value` (non-empty string)
- `listingOrder`
    - Can be implicit from the serialization.

Children:

- `Usage` (zero or more)
- `Pronunciation` (zero or more)

Example 9. Example of an entry with two inflected forms

```
LexicographicResource: language="ga", displayLanguage="en"
    PartOfSpeechCaption: value="nf", displayValue="feminine noun"
    InflectedFormCaption: value="gs", displayValue="genitive singular"
    InflectedFormCaption: value="pl", displayValue="plural"
    Entry headword="deirfiúr"
        PartOfSpeech: value="nf"
        InflectedForm: caption="gs", value="deirféar"
        InflectedForm: caption="pl", value="deirfiúracha"
```

3.1.8 `Example`

A sentence or other text extract which illustrates the headword being used in the sense.

Child of:

- `Sense`

Properties:

- `text` (non-empty string)
- `listingOrder`
    - Can be implicit from the serialization.
- `sourceIdentity` (optional)
    - Must be one of the values defined by instances of `ExampleSource`.
- `sourceDescription` (optional, non-emty text)
    - A free-form statement about the source of the example.

Example 10. A simple example

```
LexicographicResource: language="en" displayLanguage="en"

  Entry: headword="pack"
    Sense:
      Example: text="I need to pack my bags."
```

Example 11. A sourced example

```
LexicographicResource: language="en" displayLanguage="en"
  ExampleSource value="wodehouse1967" displayValue="P.G. Wodehose (1967) The World of Jeeves. London: Arrow Books"

  Entry: headword="pack"
    Sense:
      Example: text="Have you started packing yet, Jeeves?" sourceIdentity="wodehouse1967"
```

Example 12. A sourced example with additional details about where in the source it comes from

```
LexicographicResource: language="en" displayLanguage="en"
  ExampleSource value="wodehouse1967" displayValue="P.G. Wodehose (1967) The World of Jeeves. London: Arrow Books"

  Entry: headword="pack"
    Sense:
      Example: text="Have you started packing yet, Jeeves?" sourceIdentity="wodehouse1967" sourceDescription="p. 314 line 20"
```

Example 13. An example with ad-hoc attribution

```
LexicographicResource: language="en" displayLanguage="en"

  Entry: headword="pack"
    Sense:
      Example: text="Pack up your knives and go!" sourceDescription="overheard in Dublin corner of Grafton St. and Stephen's Green"
```

3.2 Metadata types

3.2.1 `InflectedFormCaption`

An `InflectedFormCaption` represents one of several allowed values for the `caption` property of `InflectedForm` objects.

Properties:

- `value` (non-empty string)

- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

Example 14. Example: restricting the captions of inflected forms

```
LexicographicResource: language="ga", displayLanguage="en"
    PartOfSpeechCaption: value="nf", displayValue="feminine noun"
    InflectedFormCaption: value="gs", displayValue="genitive singular"
    InflectedFormCaption: value="pl", displayValue="plural"
    Entry headword="deirfiúr"
        PartOfSpeech: value="nf"
        InflectedForm: caption="gs", value="deirféar"
        InflectedForm: caption="pl", value="deirfiúracha"
```

Example 15. Example: mapping the captions of inflected forms to external inventories

```
InflectedFormCaption: value="gs", displayValue="genitive singular"
    Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#singular"
    Mapping: sameAs= "http://www.lexinfo.net/ontology/3.0/lexinfo#genitiveCase"
InflectedFormCaption: value="pl", displayValue="plural"
    Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#plural"
```

3.2.2 `PartOfSpeechValue`

Represents one of several allowed values for the `value` property of `PartOfSpeech` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

Example 16. Example: restricting part-of-speech values

```
LexicographicResource: displayLanguage="en"
    PartOfSpeechValue: value="n", displayValue="noun"
    PartOfSpeechValue: value="v", displayValue="verb"
    PartOfSpeechValue: value="adj", displayValue= "adjective"
    Entry: headword="aardvark"
        PartOfSpeech: value="n"
    Entry: headword="speak"
        PartOfSpeech: value="v"
    Entry: headword="small"
        PartOfSpeech: value="adj"
```

Example 17. Example: mapping part-of-speech values to external inventories

```
PartOfSpeechValue: value="n", displayValue="noun"
    Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#noun"
```

Example 18. Example: mapping complex part-of-speech values to external inventories

```
PartOfSpeechValue: value="Nma", displayValue="masculine animate noun"
    Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#noun"
    Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#masculine"
    Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#animate"
```

3.2.3 `LabelValue`

Represents one of several allowed values for the `value` property of `Label` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

Example 19. Example: restricting the values of Label objects

```
LexicographicResource: displayLanguage="en"
    LabelValue: value="sl", displayValue="slang"
    LabelValue: value="vul", displayValue="vulgar"
```

Example 20. Example: mapping the values of Label objects to external ontologies

```
LexicographicResource: displayLanguage="en"
    LabelValue: value="sl", displayValue="slang"
        Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#slangRegister"
    LabelValue: value="vul", displayValue="vulgar"
        Mapping: sameAs="http://www.lexinfo.net/ontology/3.0/lexinfo#vulgarRegister"
```

3.2.4 `ExampleSource`

Represents one of several allowed sources for the `sourceIdentity` property of `Example` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

3.2.5 `Mapping`

Represents the fact that an item in the controlled vocabulary is equivalent to item provided by en external authority.

Parents:

- `PartOfSpeechCaption`
- `TranslationPartOfSpeechCaption`
- `LabelCaption`
- `EntrySetCaption`
- `SenseSetCaption`
- `SensePairCaption`
- `SenseTupleCaption`
- `TranslationLabelCaption`
- `InflectedFormCaption`
- `TranslationInflectedFormCaption`

Properties:

- `sameAs` (URI)

4 Bilingual Module

Extends DMLex Core to support the encoding of bilingual lexicographic resources.

4.1 Extensions to core data types

4.1.1 `LexicographicResource`

Additional properties:

- `translationLanguage` (optional, IETF language code)
- `translationTranscriptionScheme` (optional, reference to some external authority - which?)
  - The scheme (e.g. IPA) in which the `transcription` property of `TranslationPronunciation` objects is given.

4.1.2 `Sense`

Additional children:

- `HeadwordTranslation` (zero or more)

4.1.3 `Example`

Additional children:

- `ExampleTranslation` (zero or more)

4.2 Data types

4.2.1 `HeadwordTranslation`

The translation equivalent of the headword in one of its senses.

Child of:

- `Sense`

Properties:

- `text` (non-empty string)
  - Can be a single word, a multi-word expression, or indeed any expression in the target language.
- `listingOrder`
  - Can be implicit from the serialization.

Children:

- `TranslationPartOfSpeech` (zero or more)
- `TranslationLabel` (zero or more)

- `TranslationPronunciation` (zero or more)
- `TranslationInflectedForm` (zero or more)

4.2.2 `TranslationPartOfSpeech`

Any of the word classes to which the translation may be assigned, e.g. noun, verb, adjective, etc.

Child of:

- `HeadwordTranslation`

Properties:

- `value` (non-empty string)
  - Must be one of the values defined by instances of `TranslationPartOfSpeechValue`.
- `listingOrder`
  - Can be implicit from the serialization.

4.2.3 `TranslationLabel`

An indication of some restriction on the use of its parent. The restriction can be pragmatic (time, region, register), semantic (domain, semantic type) or formal ('no plural').

Child of:

- `HeadwordTranslation`
- `TranslationPronunciation`
- `TranslationInflectedForm`

Properties:

- `value` (non-empty string)
  - Must be one of the values defined by instances of `TranslaionLabelValue`.
- `listingOrder`
  - Can be implicit from the serialization.

4.2.4 `TranslationPronunciation`

Information about the pronunciation of its parent.

Child of:

- `HeadwordTranslation`
- `TranslationInflectedForm`

Properties (at least one):

- `transcription` (non-empty string)
- `recording` (string: name or URL of a sound file)
- `listingOrder`
  - Can be implicit from the serialization.

Children:

- `TranslationLabel` (zero or more)

4.2.5 `TranslationInflectedForm`

A form of the inflectional paradigm of its parent.

Child of:

- `HeadwordTranslation`

Properties:

- `label` (non-empty string) e.g. 'plural'
  - Must be one of the values defined by instances of `TranslationInlectedFormCaption`.
- `value` (non-empty string)
- `listingOrder`
  - Can be implicit from the serialization.

Children:

- `TranslationLabel` (zero or more)
- `TranslationPronunciation` (zero or more)

4.2.6 `HeadwordTranslation`

The translation equivalent of the headword in one of its senses.

Child of:

- `Sense`

Properties:

- `text` (non-empty string)
    - Can be a single word, a multi-word expression, or indeed any expression in the target language.
- `listingOrder`
    - Can be implicit from the serialization.

Children:

- `TranslationPartOfSpeech` (zero or more)
- `TranslationLabel` (zero or more)
- `TranslationPronunciation` (zero or more)
- `TranslationInflectedForm` (zero or more)

4.2.7 `ExampleTranslation`

The translation of an example.

Child of:

- `Example`

Properties:

- `text` (non-empty string)
- `listingOrder`
    - Can be implicit from the serialization.

4.3 Metadata types

4.3.1 `TranslationInflectedFormCaption`

Represents one of several allowed values for the `caption` property of `TranslationInflectedForm` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

4.3.2 `TranslationLabelValue`

Represents one of several allowed values for the `value` property of `TranslationLabel` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

4.3.3 `TranslationPartOfSpeechValue`

Represents one of several allowed values for the `value` property of `TranslationPartOfSpeech` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

5 Internal Linking Module

Used to construct clickable (or otherwise navigable) hyperlinks between objects within the same `LexicographicResource`.

5.1 Extensions to core data types

5.1.1 `LexicographicResource`

Additional children:

- `EntrySetCaption` (zero or more)
- `SenseSetCaption` (zero or more)
- `SensePairCaption` (zero or more)
- `SenseTupleCaption` (zero or more)
- `SenseEntryTupleCaption` (zero or more)

5.1.2 `Entry`

Additional properties:

- `id`

5.1.3 `Sense`

Additional properties:

- `id`

5.2 Relation types

5.2.1 `EntrySet`

Represents an unordered set of two or more entries. Recommended for encoding variants such for example masculine and feminine counterparts or spelling variants.

Properties:

- `caption` (non-empty string) eg. "variants"
    - Must be one of the values defined by instances of `EntrySetCaption`.

Participants (all in the same `LexicographicResource`):

- `entry` (two or more)
    - Reference to an `Entry` object.

Example 21. Encoding variants

```
LexicographicResource: language="en", displayLanguage="en"
    LabelValue: value="eu", displayValue="European English"
    LabelValue: value="am", displayValue="American English"
    EntrySetCaption value="vars" displayValue="variants"

    Entry id="#colour" headword="colour"
        Label: value="eu"

    Entry id="#color" headword="color"
        Label: value="am"

    EntrySet caption="vars"
        entry: #colour
        entry: #color
```

Example 22. Encoding gender pairs

```
LexicographicResource: language="en", displayLanguage="en"
    EntrySetCaption value="gpair" displayValue="gender pair"

    Entry id="#actor" headword="actor"

    Entry id="#actress" headword="actress"

    EntrySet caption="gpair"
        entry: #actor
        entry: #actress
```

Example 23. Encoding homonyms

```
LexicographicResource: language="de" displayLanguage="en" translationLanguage="en"
    EntrySetCaption value="homo" displayValue="homonyms"
    PartOfSpeechValue value="n-masc" displayValue="masculine noun"
    PartOfSpeechValue value="n-fem" displayValue="feminine noun"

    Entry id="#der-see" headword="See"
        PartOfSpeech: value="n-masc"
        Sense: id="#der-see-1"
                Translation: lake

    Entry id="#die-see" headword="See"
        PartOfSpeech: value="n-fem"
        Sense: id="#die-see-1"
                Translation: see

    EntrySet caption="homo"
```

```
        entry: #der-see
        entry: #die-see
```

5.2.2 `SenseSet`

Represents an unordered set of two or more senses (typically - but not necessarily - belonging to two or more different entries). Recommended for encoding synonyms, near synonyms etc.

Properties:

- `caption` (non-empty string) eg. "synonyms"
  - Must be one of the values defined by instances of `SenseSetCaption`.

Participants (all in the same `LexicographicResource`):

- `sense` (two or more)
  - Reference to a `Sense` object.

Example 24. Encoding near synonyms

```
LexicographicResource: language="de" displayLanguage="en" translationLanguage="en"
    EntrySetCaption value="nsyn" displayValue="near synonyms"
    PartOfSpeechValue value="n-masc" displayValue="masculine noun"
    PartOfSpeechValue value="n-fem" displayValue="feminine noun"
    PartOfSpeechValue value="n-neut" displayValue="neuter noun"

    Entry id="#die-see" headword="See"
       PartOfSpeech: value="n-fem"
        Sense: id="#die-see-1"
            Translation: "see"

    Entry id="#das-meer" headword="Meer"
       PartOfSpeech: value="n-neut"
        Sense: id="#das-meer-1"
            Translation: "see"

    Entry id="#der-ozean" headword="Ozean"
       PartOfSpeech: value="n-masc"
        Sense: id="#der-ozean-1"
            Translation: "ocean"

    SenseSet caption="nsyn"
        sense: #die-see-1
        sense: #das-meer-1
        sense: #der-ozean-1
```

5.2.3 `SensePair`

Represents an undirected pair of two senses (typically - but not necessarily - belonging to two different entries). Recommended for representing pairs of antonyms, opposites, converses and so on.

Properties:

- `caption` (non-empty string) eg. "antonyms"
  - Must be one of the values defined by instances of `SensePairCaption`.

Participants (all in the same `LexicographicResource`):

- `sense` (exacly two)
  - Reference to a `Sense` object.

Example 25. Encoding antonyms

```
LexicographicResource: language="en" displayLanguage="en"
    EntrySetCaption value="anto" displayValue="antonyms"

    Entry id="#buy" headword="buy"
        Sense: id="#buy-1"

    Entry id="#sell" headword="sell"
        Sense: id="#see-1"

    SensePair caption="anto"
        entry: #buy-1
        entry: #sell-1
```

5.2.4 `SenseTuple`

Represents a directed pair of two senses. When the two senses belong to two different entries, then this type is recommended for representing pairs of hypernym/hyponym, broader/narrower, holonym/meronym. When the two senses belong to the same entry, then this type is recommended for representing the relationship between a sense and a subsense.

Properties:

- `caption` (non-empty string) eg. "hypernymy"
  - Must be one of the values defined by instances of `SenseTupleCaption`.

Participants (all in the same `LexicographicResource`):

- `superordinate` (exacly one)
  - Reference to a `Sense` object.

- `subordinate` (exacly one)
    - Reference to a `Sense` object.

Example 26. Encoding meronymy

```
LexicographicResource: language="en" displayLanguage="en"
  EntrySetCaption value="mero" displayValue="meronymy"

  Entry id="#glasses" headword="glasses"
    Sense: id="#glasses-1"

  Entry id="#microscope" headword="microscope"
    Sense: id="#microscope-1"

  Entry id="#lense" headword="lense"
    Sense: id="#lense-1"

  SenseTuple caption="mero"
    superordinate: #glasses-1
    subordinate: #lense-1

  SenseTuple caption="mero"
    superordinate: #microscope-1
    subordinate: #lense-1
```

5.2.5 `SenseEntryTuple`

Represents a directed pair of a sense and an entry, such that the entry does *not* contain the sense. This type is recommended to be used for situations when the entry is expected to be show to a human user as a subentry inside the sense of another entry.

Properties:

- `caption` (non-empty string) eg. "subentry"
    - Must be one of the values defined by instances of `SenseEntryTupleCaption`.

Participants (all in the same `LexicographicResource`):

- `superordinate` (exacly one)
    - Reference to a `Sense` object.
- `subordinate` (exacly one)
    - Reference to an `Entry` object.

Example 27. Example: encoding a subentry at subsense level

In this example, when the entry for safe is being shown to a human user, the entry for better safe than sorry is supposed to appear as a subentry at at the end of the first sense.

```
LexicographicResource: language="en", displayLanguage="en"
  SenseEntryTupleCaption value="idiosub" displayValue="idiom subentry"

  Entry: id="#safe" headword="safe"
    Sense: id="#safe-1" indicator="protected from harm"
      Example: value="It isn't safe to park here."
    Sense: id="#safe-2" indicator="not likely to cause harm"
      Example: value="Is the ride safe for a small child?"

  Entry: id="#bsts" headword="better safe than sorry"
    Sense: definition="you should be careful even if it seems unnecessary"

  SenseEntryTuple: caption: "idiosub"
    superordinate: #safe-1
    subordinate: #bsts
```

Example 28. Example: encoding a subentry at sense level

In this example, when the entry for bible is being shown to the human user, the entry for the Bible is supposed to be appear there as a subentry, as if it were its first sense.

```
LexicographicResource: language="en", displayLanguage="en"
  SenseEntryTupleCaption value="propnamesub" displayValue="proper name subentry"

  Entry: id="#the-bible" headword="the Bible"
    Sense: id="the-bible-1" definition="the book considered holy by Christians..."

  Entry: id="#bible" headword="bible"
    Sense: id="#bible-1" //empty sense, serves as a container for "the Bible"
    Sense: id="#bible-2" definition="a book considered important..."

  SenseEntryTuple: caption="propnamesub"
    superordinate: #bible-1
    subordinate: #the-bible
```

5.3 Metadata types

5.3.1 `EntrySetCaption`

Represents one of several allowed values for the `caption` property of `EntrySet` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

5.3.2 `SenseSetCaption`

Represents one of several allowed values for the `caption` property of `SenseSet` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

5.3.3 `SensePairCaption`

Represents one of several allowed values for the `caption` property of `SensePair` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

5.3.4 `SenseTupleCaption`

Represents one of several allowed values for the `caption` property of `SenseTuple` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

5.3.5 `SenseEntryTupleCaption`

Represents one of several allowed values for the `caption` property of `SenseEntryTuple` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `Mapping` (zero or more)

6 External Linking Module

Used to construct links between objects from different lexicographic resources.

6.1 Relation types

6.1.1 `Same`

Represents the fact that two or more senses from two or more lexicographic resources are semantically the same.

Participants:

- `Sense` (two or more, no two of which belong to the same `LexicographicResource`)

6.1.2 `Subsumption`

Represents the fact that one sense from one lexicographic resource is semantically wholly contained inside another sense from another lexicographic resource.

Participants:

- the broader `Sense` (exactly one)
- the narrower `Sense` (exactly one)

The two senses must not be within the `LexicographicResource`.

6.1.3 `Overlap`

Represents the fact that two or more senses from two or more lexicographic resources overlap semantically, without either one being semantically wholly contained inside the other.

Participants:

- `Sense` (two or more, no two of which belong to the same `LexicographicResource`)

7 Inline Markup Module

7.1 Marker types

7.1.1 `PlaceholderMarker`

Marks up a substring inside a headword (or inside a headword translation) which is not part of the expression itself but stands for things that can take its place, or constitutes some kind of meta-notation. Examples:

- `beat [sb.] up`
- `continue [your] studies`

Markup of:

- `headword` property of `Entry`
- `text` property of `HeadwordTranslation`

7.1.2 `HeadwordMarker`

Marks up a substring inside an example (or inside an example translation) which corresponds to the headword (or to a translation of the headword).

Markup of:

- `text` property of `Example`
- `text` property of `ExampleTranslation`

7.1.3 `CollocateMarker`

Marks up a substring inside an example (or inside an example translation) where an important collocate of the headword (or of its translation) occurs.

Markup of:

- `text` property of `Example`
- `text` property of `ExampleTranslation`

Properties:

- `lemma`
    - Optional. The lemmatized form of the collocate.

Example 29. Marking up the headword and a collocate inside an example

```
Example: value="The coroner performed an autopsy."
  HeadwordMarker: value=first occurence of "autopsy"
  CollocateMarker: value=first occurence of "performed" lemma="perform"
```