Proceedings of the Globalex Workshop on Linked Lexicography @LREC2022, pages 10–18
Marseille, 20 June 2022

Modelling Collocations in OntoLex-FrAC

Christian Chiarcos[1,2], Katerina Gkirtzou[3], Maxim Ionov[1,2], Besim Kabashi[4], Anas Fahad Khan[5], Ciprian-Octavian Truică[6]
[1]Applied Computational Linguistics, Goethe University Frankfurt, Frankfurt am Main, Germany
[2]Institute for Digital Humanities, University of Cologne, Germany
[3] Institute of Language and Speech Processing, Athena Research Center, Athens, Greece
[4]Computational and Corpus Linguistics, Friedrich-Alexander University of Erlangen-Nuremberg, Germany
[5]Istituto di Linguistica Computazionale A. Zampolli, Consiglio Nazionale delle Ricerche, Italy
[6]Department of Information Technology, Uppsala University, Sweden
[1]{chiarcos,ionov}@cs.uni-frankfurt.de, [3]katerina.gkirtzou@athenarc.gr, [4]besim.kabashi@fau.de,
[5]fahad.khan@ilc.cnr.it, [6]ciprian-octavian.truica@it.uu.se

Abstract

Following presentations of frequency and attestations, and embeddings and distributional similarity, this paper introduces the third cornerstone of the emerging OntoLex module for Frequency, Attestation and Corpus-based Information, OntoLex-FrAC. We provide an RDF vocabulary for collocations, established as a consensus over contributions from five different institutions and numerous data sets, with the goal of eliciting feedback from reviewers, workshop audience and the scientific community in preparation of the final consolidation of the OntoLex-FrAC module, whose publication as a W3C community report is foreseen for the end of this year. The novel collocation component of OntoLex-FrAC is described in application to a lexicographic resource and corpus-based collocation scores available from the web, and finally, we demonstrate the capability and genericity of the model by showing how to retrieve and aggregate collocation information by means of SPARQL, and its export to a tabular format, so that it can be easily processed in downstream applications.

Keywords: lexical resources, standards, OntoLex, collocation analysis

1. Background

Since its publication in 2016, the OntoLex-Lemon vocabulary (McCrae et al., 2017) has become the dominant vocabulary for modelling machine-readable dictionaries on the Semantic Web. OntoLex-FrAC, the OntoLex module for *Frequency, Attestation and Corpus information*, is an emerging vocabulary for enriching machine-readable lexicons with corpus information. Since 2018, OntoLex-FrAC has been under development as a companion vocabulary for (and a module of) OntoLex-Lemon in the context of the W3C community group Ontology-Lexica (OntoLex). The module is targeted at complementing dictionaries and other linguistic resources containing lexicographic data with a vocabulary to express the lexical information found in or derived from corpora, i.e., (collections of) text, written or spoken.

The current OntoLex-FrAC vocabulary is illustrated in Fig. 1. Previous publications discussing OntoLex-FrAC centered on attestations and frequency (Chiarcos et al., 2020) and corpus-based information such as embeddings and distributional similarity (Chiarcos et al., 2021). Here, we describe the extension of OntoLex-FrAC for collocation analysis.

In linguistics, the term *collocation* is used to describe the analysis of word combinations. Many groups of words can be freely combined with each other, whereas others have a strong tendency to co-occur, while others can only be combined with a limited number of other words, or are even part of fixed idioms. For example, English *heavy rain* is a common phrase, whereas *strong rain* is not. But this is language-specific: German *starker Regen* ("strong rain") is common while *schwerer Regen* ("heavy rain") is not.

The analysis of collocations and their automated retrieval from corpora is a key technique in modern digital lexicography: It supports lexicographers in identifying context-dependent patterns of use of a particular lexeme, which can then stimulate and direct further lexicographic analysis. A number of tools for this purpose have been developed, e.g., SketchEngine (Kilgarriff et al., 2014) and Corpus WorkBench (Hardie, 2012), and although they currently lack machine-readable interface specifications, their APIs represent a de facto standard in digital lexicography. OntoLex-FrAC is dedicated to addressing this gap and closely follows the requirements of these tools. At the same time, collocation dictionaries are also lexicographic resources in their own right, e.g., as tools to support learners and second language speakers in finding contextually appropriate expressions, and they have characteristics that set them apart from both general-purpose machine-readable dictionaries (covered by OntoLex-Lemon) and traditional dictionaries as used and created in lexicographic research (covered by OntoLex-Lexicog, Bosque-Gil and Gracia, 2019). OntoLex-FrAC covers both use cases: collocation dictionaries and automated collocation analysis.

Within OntoLex, collocations have been modeled for the first time as part of **OntoLex-FrAC**, and to the

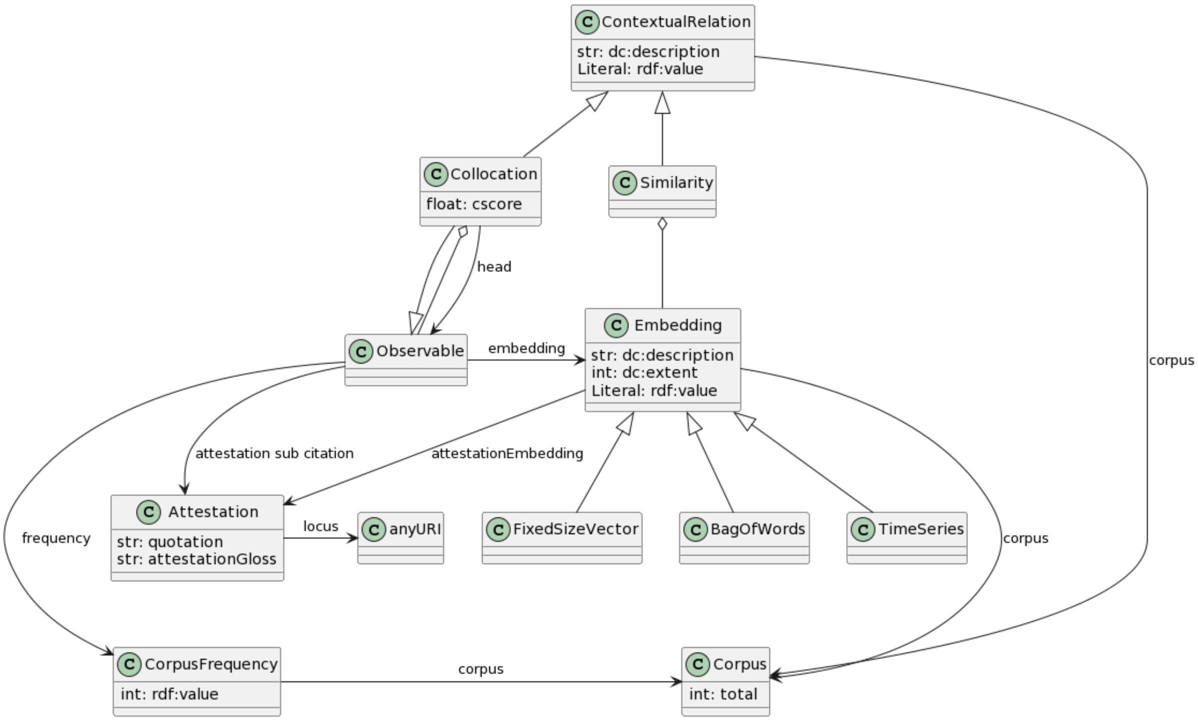


Figure 1: OntoLex-FrAC, draft version of March 2022 as UML class diagram, cf. Suchánek and Pergl (2020) for notational conventions

best of our knowledge, no machine-readable vocabulary for collocation dictionaries and related resources on the basis of RDF technologies has been suggested before. Some precedent may be seen in the collocation vocabulary for lexical entries as described in the XML-based Text Encoding Initiative (TEI) guidelines (Initiative, 2022). Although TEI is not Linked Data based, it does give us a useful point of reference for seeing how collocations can be representing as structured data in computational lexicons.

In fact, there are at least three different ways of representing collocations in TEI lexicons, using different vocabulary elements, one being `colloc` (‘sequence of words that co-occur with the headword with significant frequency’)[1]. Secondly, collocations can also be specified using the `gram` element (as part of the grammatical description of a lexical entry), as is seen in the example given of the preposition *de* collocate of the French word *médire* given in Section 9.3.2 of the TEI guidelines. Thirdly, collocations can be described using the usage element `usg` by specifying the `@type` attribute of the element as ”colloc”. The important insights to be drawn from the TEI guidelines is that (a) there is a demand for modelling collocations in the context of dictionaries (hence multiple, incompatible ways to model it, driven by different use cases and requirements), but that (b) at the moment, the support for modelling collocation scores in this context is severely limited. From the options mentioned above only `colloc` allows to specify collocation scores by adding a `certainty` element and *ab*using its `@cert` attribute, which, however, is only used with human-readable labels in the guidelines,[2] but neither with numerical scores nor with a systematic means of defining the type of collocation score.

2. Collocations in OntoLex-FrAC

The base element of OntoLex-FrAC is `frac:Observable`, i.e., any element that observations can be made about *in a corpus*. This corpus-based focus also defines our understanding of collocations not as lexical units, but as being characterized by certain association scores (for which high values may hint at a lexicalized collocation, but which can be calculated and returned for *any* combination of words). Typical observables are words (`ontolex:Form`) or lexemes (`ontolex:LexicalEntry`), but also lexical concepts or general ontological concepts can be observed – if annotated in a corpus. This definition of observables – motivated from other aspects of corpus-based information before – is organically applicable to collocation analysis: collocations are usually defined on surface-oriented criteria, i.e., as a relation between forms or lemmas (lexical entries), not between senses, but they can be analyzed on the level of word senses (the sense that gave rise to the idiom or collocation).

Collocations are not constrained to pairs of words, longer collocations are also possible. Accordingly, we

[1] `https://tei-c.org/release/doc/tei-p5-doc/en/html/ref-colloc.html`

[2] `https://tei-c.org/release/doc/tei-p5-doc/en/html/ref-certainty.html`

model collocations as an aggregate of observables, not as a relation between words. Moreover, collocations *are* observables in their own right. In particular, they can have attestations (i.e., corpus examples that show the words under consideration in context, frequencies, similarity scores, etc.).

Collocations obtained by quantitative methods are :

Def. 2.1 (`frac:Collocation`)**.** An RDF container (`rdfs:Container`, i.e., `rdf:Seq` or `rdf:Bag`) that contains two or more `frac: Observables` based on their co-occurrence within the same context window and that can be characterized by their method of creation (`dct: description`), their collocation score (weight, collocation strength) (`frac:cscore`), and the corpus used to create them (`frac:corpus`).

Collocations may have fixed or variable word order. Where fixed word order is required, the collocation must be defined as a sequence (`rdf:Seq`), otherwise, the default interpretation is as an unordered set (`rdf:Bag`). The elements of any collocation can be accessed by `rdfs:member`. Optionally, the elements of an ordered collocation can be accessed by numerical indices (`rdf:_1`, `rdf:_2`, etc.).

Additional parameters such as the size of the context window used for collocation analysis can be provided in human-readable form in `dct:description`.

Note that FrAC collocations can be used to represent collocations both in the lexicographic sense (as complex units of meaning) and in the quantative sense (as determined by collocation metrics over a particular corpus), but that the quantitative interpretation is the preferred one in the context of FrAC. To mark collocations in the lexicographic sense as such, they can be assigned a corresponding `lexinfo:termType`, e.g., by means of `lexinfo:idiom`, `lexinfo: phraseologicalUnit` or `lexinfo:set Phrase`. If explicit sense information is being provided, the recommended modelling is by means of `ontolex:MultiWordExpression`; it can be defined as `frac:Collocation` (`rdfs:member` can be left implicit).

In automated collocation analysis, collocations can be described in terms of various collocation scores:

Def. 2.2 (`frac:cscore`)**.** Collocation score is a subproperty of `rdf:value` that provides the value for one specific type of collocation score for a particular collocation in its respective corpus.

We define popular collocation metrics as sub-properties of `frac:cscore` (Sect. 3). For those that are asymmetric (e.g., `frac:relFreq`), we distinguish the lexical element they are about (the head) from its collocate(s). If such metrics are provided, a collocation should identify the element that it conveys information about, modelled here with the property `frac:head`:

Def. 2.3 (`frac:head`)**.** Identifies the `rdfs: member` of a collocation that its scores are about. A collocation must not have more than one head.

3. Collocation Scores

OntoLex-FrAC defines popular collocation scores as sub-properties of `frac:cscore`, and users are encouraged to define their own subproperties if different scores are being used. In case only one kind of score is provided by a source, users can also use `rdf:value` along with a `dct:description` explaining the metric. We present selected sub-properties along with their mathematical definition.

Def. 3.1 (`frac:relFreq`)**.** Relative frequency indicates how often a specific word y in the collocation occurs together with the head word x: $\texttt{relFreq}_x = \frac{p(x,y)}{p(x)}$.

Def. 3.2 (`frac:pmi`)**.** Pointwise Mutual Information (PMI) measures the extent to which the words in a collocation occur more frequently than by chance. If two words appear together more than expected under independence there must be some kind of semantic relationship between them (Role and Nadif, 2011). Thus, PMI is the log of the ratio of the observed co-occurrence frequency to the frequency expected under independence: $\text{PMI}(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$

PMI variants, such as normalized PMI, cf. (Role and Nadif, 2011), are provided as well, i.e. `frac:npmi`, `frac:pmi2` and `frac:pmi3`.

Def. 3.3 (`frac:dice`)**.** Dice coefficient is a statistic used to gauge the collocation of two words x and y (Manning and Schutze, 1999): $\texttt{dice}(x, y) = \frac{2p(x,y)}{p(x)+p(y)}$

Def. 3.4 (`frac:minSensitivity`)**.** Minimum sensitivity is computed as the minimum between the relative sensitivity of word x and of word y (Pedersen, 1998): $\texttt{minSensitivity}(x, y) = \min(\frac{p(x,y)}{p(y)}, \frac{p(x,y)}{p(x)})$

In addition to collocation scores, statistical independence tests are employed as collocation scores, including `frac:tScore` (Student's t test), `frac:chi2` (Pearson's χ^2), `frac:likelihood_ratio` (Log Likelihood Ratio test) (Manning and Schutze, 1999).

Furthermore, related metrics from disciplines other than computational lexicography and corpus linguistics are also provided as `frac:cscore` subproperties. In association rule mining, for example, an association rule $x \rightarrow y$ corresponds to a collocation in that the existence of word x implies the existence of word y.

Def. 3.5 (`frac:support`)**.** indicates how frequently the rule appears in the dataset (Larose and Larose, 2014): $\texttt{support}(x \rightarrow y) = p(x, y)$

Def. 3.6 (`frac:confidence`)**.** indicates how often the rule has been found to be true (Larose and Larose, 2014): $\texttt{confidence}(x \rightarrow y) = \frac{p(x,y)}{p(x)}$

Def. 3.7 (`frac:lift`)**.** (or interest of a rule) measures how many times more often x and y occur together than expected if they are statistically independent (Larose and Larose, 2014): $\texttt{lift}(x \rightarrow y) = \frac{p(x,y)}{p(x)p(y)}$

Def. 3.8 (`frac:conviction`)**.** (conviction of a rule) is the ratio of the expected frequency that x occurs without y, i.e., the frequency that the rule makes an incorrect prediction, if x and y are independent divided by the observed frequency of incorrect predictions (Brin et al., 1997): conviction$(x \rightarrow y) = \frac{p(x)p(\neg y)}{p(x,\neg y)}$

Where:

- x, y - the (head) of the word and its collocate
- $p(x)$, $p(y)$ the probabilities of word x and y
- $p(\neg x) = 1 - p(x)$
- $p(x, y)$ the probability of the co-occurrence of x and y

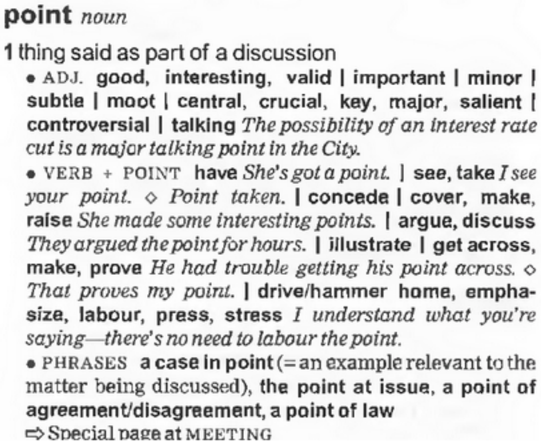
4. Case Studies

We illustrate the application of OntoLex-FrAC to (a) the conversion of an existing collocation dictionary to a machine-readable format, and (b) its enrichment with collocation scores obtained from an external corpus. It is to be noted, however, that OntoLex-FrAC is not an independent vocabulary, but that it builds on OntoLex (and can thus complement existing OntoLex data). It can also be applied in conjunction with other OntoLex modules. We illustrate the conjoined application of OntoLex-FrAC and OntoLex-Lexicog to the Oxford Collocation Dictionary for Students.

4.1. The Oxford Collocations Dictionary

We show an example of the application of Ontolex-FrAC by looking at an example encoding of the entry for the word *point* from *the Oxford Collocations Dictionary for Students of English* (OCDS) (OUP, 2002). Figure 2 shows how the OCDS groups together the entry with individual collocations for better accessibility and readability.

For instance *point*-collocations are first grouped together on the sense level, then on the basis of the part of speech of the collocated word and/or whether the collocation constitutes a phrase, and finally at the level of similarity of meaning of the collocation (note that there is also a division of examples for the same meaning grouping). In the OCDS the separation of groupings on the basis of meaning is visually effected by the | symbol. We refer to these (potentially nested) groupings of collocation information as *collocation patterns* in what follows. The *point* example is interesting for showing how OntoLex-FrAC can be used together with the OntoLex-Lexicographic model.



point *noun*

1 thing said as part of a discussion

- ADJ. **good, interesting, valid | important | minor | subtle | moot | central, crucial, key, major, salient | controversial | talking** *The possibility of an interest rate cut is a major talking point in the City.*
- VERB + POINT **have** *She's got a point.* | **see, take** *I see your point.* ◇ *Point taken.* | **concede | cover, make, raise** *She made some interesting points.* | **argue, discuss** *They argued the point for hours.* | **illustrate | get across, make, prove** *He had trouble getting his point across.* ◇ *That proves my point.* | **drive/hammer home, emphasize, labour, press, stress** *I understand what you're saying—there's no need to labour the point.*
- PHRASES **a case in point** (= an example relevant to the matter being discussed), **the point at issue, a point of agreement/disagreement, a point of law**

⇨ Special page at MEETING

Figure 2: Entry for *point* in the Oxford Collocations Dictionary

Note that in our RDF modelling we represent the collocations themselves using the FrAC vocabulary and the domain-specific segmentation of the entry into collocation patterns using OntoLex-Lexicog. Indeed we use the class `lexicog:LexicographicComponent` to represent this organisation that is so typical of collocation dictionaries.

We start by looking at the modelling of the lexical content of the entry and introduce the `:point` lexical entry, giving part of speech information about the word and about its lemma form. We also introduce `:ls_point_1`, the first sense of the word corresponding to the first sense listed in the dictionary entry in Figure 2 (we only look at this first sense in the following example).

```
:point a ontolex:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:noun ;
  ontolex:sense :ls_point_1 ;
  ontolex:canonicalForm
    [ ontolex:writtenRep "point"] .

:ls_point_1 a ontolex:LexicalSense ;
    # p_s
    skos:definition "thing said as part
        of a discussion" .
```

The following lexical entries represent the collocates of the word *point*. We will refer to these entries in the descriptions of the collocations below:

```
:have a ontolex:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:verb ;
  ontolex:canonicalForm
    [ ontolex:writtenRep "have"] .

:see a ontolex:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:verb ;
  ontolex:canonicalForm
    [ ontolex:writtenRep "see" ] .

:take a ontolex:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:verb ;
  ontolex:canonicalForm
    [ ontolex:writtenRep "take" ] .
```

The collocations of *point*, or to be more accurate the collocations of the first sense of the word *point*, are represented using the FrAC classes which we introduce as follows.

```
:col_have_point a frac:Collocation ,
    rdf:Seq ;
  lexinfo:example "She's got a point" ;
  frac:head :ls_point_1 ;
  rdf:_1 :have ;
  rdf:_2 :ls_point_1 .

:col_see_point a frac:Collocation ,
    rdf:Seq ;
  lexinfo:example "I see your point" ;
  frac:head :ls_point_1 ;
  rdf:_1 :see ;
  rdf:_2 :ls_point_1 .

:col_take_point a frac:Collocation ,
    rdf:Seq ;
  lexinfo:example "Point taken" ;
  frac:head :ls_point_1 ;
  rdf:_1 :take ;
  rdf:_2 :ls_point_1 .
```

Note the use of the property head to specify the head of the collocation in each case, as well as that of the lexinfo property example to give the example presented in the original entry. Note in addition the use of rdf:_1 and rdf:_2 to represent the order of the collocates.

Next we represent the arrangement of this information as it is found in the dictionary itself using lexicog classes and lexicog:LexicographicComponent in particular. The dictionary entry (as opposed to the lexical entry) for *point* is represented by :e_point an individual of type lexicog:Entry. As we can see below, :e_point is linked to the lexical entry :point via the lexicog:describes property.

```
:e_point a lexicog:Entry ;
  lexicog:describes :point ;
  lexicog:subComponent
  [  a lexicog:LexicographicComponent ;
    lexicog:describes :ls_point_1 ;
    lexicog:subComponent
      :lc_point_pattern_1 ,
      :lc_point_pattern_2 ] .
```

For reasons of space we only (partially) model two of the collocation patterns in the entry in our RDF encoding: those pertaining to the collocation of the word *point* with an adjective and those pertaining to its collocation with a proceeding verb. These are :lc_point_pattern_1 and :lc_point_pattern_2 respectively. Both of these are lexicog lexicographic components. The text associated with each in the original entry is specified using the property dct: description.

```
:lc_point_pattern_1
  a lexicog:LexicographicComponent ;
  dct:description "ADJ" .

:lc_point_pattern_2
  a lexicog:LexicographicComponent ;
  dct:description "VERB + POINT" ;
  lexicog:subComponent :lc_have_point ,
    :lc_see_take_point .
```

Note that :lc_point_pattern_2 is broken up into two further collocation patterns; the first, :lc_have_point, describes the word's collocates with *have*, and the second, :lc_see_take_point, its collocates with *see* and *take*. These are described below.

```
:lc_have_point
  a lexicog:LexicographicComponent ;
  lexicog:describes :col_have_point .
:lc_see_take_point
  a lexicog:LexicographicComponent ;
  lexicog:describes :col_see_point ,
        :col_take_point .
```

4.2. Enrichment with Collocation Scores

Aside from lexicographic expertise, the ODCS builds on (but does not provide) collocation scores. However, these can be added from other sources. One example here is the Leipzig Corpora Collection / Deutscher Wortschatz, a project of Leipzig University, the Saxon Academy of Sciences and Humanities in Leipzig and the Institute for Applied Informatics (Goldhahn et al., 2012).

Considering the word *point* in the English News (2020) corpus at the Wortschatz portal,[3] we find that *see* co-occurs with *point* 544 times (co-occurrence in the same sentence), while *point* occurs 183,306 times. In OntoLex-FrAC, the absolute frequencies can be modelled as follows:

```
:N2020_Frequency
  rdfs:subClassOf frac:CorpusFrequency,
    [ a owl:Restriction ;
      owl:onProperty frac:corpus ;
      owl:hasValue
        <https://corpora.uni-leipzig.de/en/
            res?corpusId=eng_news_2020>
    ] .

:col_see_point
  frac:frequency
```

[3] https://corpora.uni-leipzig.de/en/res?corpusId=eng_news_2020&word=point

```
    [ a :N2020_Frequency ;
      rdf:value "544" ] .

:point
  frac:frequency
    [ a :N2020_Frequency ;
      rdf:value "183,306" ] .
```

We introduce the class `:N2020_Frequency` for frequencies from the News 2020 corpus, so that frequency declarations are compactly represented with three triples only.

The Wortschatz Portal does not provide relative frequencies, but these can be calculated, and accordingly, we can extend the original OCDS data with information such as:

```
:col_have_point
  frac:relfreq "0.002967715186628";
  frac:corpus
    <https://corpora.uni-leipzig.de/en/
    res?corpusId=eng_news_2020> .
```

It is important to note here that these scores also require to provide the original corpus URI.

5. Applications

5.1. Querying OntoLex-FrAC Data

For any downstream application of OntoLex-FrAC, queriability is the most elementary required for a user. Indeed, a key benefit of modelling lexical resources in OntoLex is that they can be processed by standard RDF tools and Linguistic Linked Open Data (LLOD) technology. Using HTTP-resolvable URIs for shared vocabularies allows to operate on consistent, well-defined and machine-readable data models, so that data can be more easily re-used. Using HTTP-resolvable URIs for the data itself allows to establish links between resources hosted by different providers, and thus to develop a decentralized ecosystem for language technology and lexical resources on the web. Over such data, the application of SPARQL includes the possibility to query across data sets hosted by different providers (SPARQL federation) and across heterogeneous data, i.e., data stored in different kinds of technical backends, be it exposed as plain files (SPARQL LOAD), via a web service (SPARQL SERVICE, e.g., an endpoint) or by means of a wrapper technology created around another kind of data source (e.g., a relational data base, using R2RML technology,[4] over XML data with GRDDL[5] or over JSON data with JSON-LD[6] context definitions).

To demonstrate the viability of our modelling for collocations, we demonstrate the application of SPARQL to retrieve data from OntoLex-FrAC from the data described in Sect. 4.1 in three different scenarios.[7]

With the first query, we retrieve all collocates per collocation:

```
SELECT DISTINCT ?collocation ?member ?order
WHERE {
  ?collocation a frac:Collocation ;
  ?prop ?member .
  FILTER(?prop=rdfs:member ||
    regex(str(?prop),".*#_[0-9]+$"))
  OPTIONAL {
    ?collocation ?nrel ?member .
    FILTER(regex(str(?nrel),".*#_[0-9]+$"))
    BIND(replace(str(?nrel),".*#_([0-9]+)$","$1")
    AS ?order )
  }
} ORDER BY ?collocation ?order ?member
```

This query evaluates two kinds of membership queries, either via `rdfs:member` (unordered) or (filter `||`) in their sequential order (if defined with `rdf:_1`, `rdf:_2`, ...). Note that with RDFS reasoning enabled at the query engine, `rdfs:member` would also be inferred from `rdf:_1`, etc.

For the ODCS sample data above, a query with Apache Jena arq retrieves the following table:

```
| collocation      | member       | order |
===========================================
| <col_have_point> | <have>       | "1"   |
| <col_have_point> | <ls_point_1> | "2"   |
| <col_see_point>  | <see>        | "1"   |
| <col_see_point>  | <ls_point_1> | "2"   |
| <col_take_point> | <take>       | "1"   |
| <col_take_point> | <ls_point_1> | "2"   |
```

The second query retrieves all collocations for a given lexical entry:

```
SELECT DISTINCT ?form ?pos
                ?collocation ?isHead
WHERE {
  ?collocation a frac:Collocation.
  ?collocation ?prop ?observable.
  FILTER(?prop=rdfs:member  ||
    regex(str(?prop),".*#_[0-9]+$"))
  ?entry
    (ontolex:sense|ontolex:lexicalForm)?
      ?observable.
  ?entry
    ontolex:canonicalForm/
    ontolex:writtenRep    ?form .
  OPTIONAL {
    ?collocation frac:head ?observable.
  BIND("true" as ?isHead)
  }
  OPTIONAL {
    ?entry lexinfo:partOfSpeech ?pos
  }
} ORDER BY ?form ?pos
    ?collocation ?isHead
```

[4] https://www.w3.org/TR/r2rml/

[5] https://www.w3.org/TR/grddl/

[6] https://www.w3.org/TR/json-ld/

[7] Queries were tested with Apache Jena 4.2.0, using the arq command line tool. For reasons of brevity, we skip prefix declarations. The following non-standard prefixes have been used: `ontolex:` `http://www.w3.org/ns/lemon/ontolex#`, `skos:` `http://www.w3.org/2004/02/skos/core#`, `frac:` `http://www.w3.org/ns/lemon/frac#`, and `lexinfo:` `http://www.lexinfo.net/ontology/3.0/lexinfo#`.

This query exploits SPARQL property paths to return collocates of any kind of observables, so the ?observable could be identical to (lexical) ?entry (no ontolex:sense or ontolex:lexicalForm relation; it could be the ontolex: sense or it could be a ontolex: lexicalForm. If defined in the data, it returns the frac:head status or the lexinfo: partOfSpeech:

```
| form    | pos          | collocation      | isHead |
=====================================================
| "have"  | lexinfo:verb | <col_have_point> |        |
| "point" | lexinfo:noun | <col_have_point> | "true" |
| "point" | lexinfo:noun | <col_see_point>  | "true" |
| "point" | lexinfo:noun | <col_take_point> | "true" |
| "see"   | lexinfo:verb | <col_see_point>  |        |
| "take"  | lexinfo:verb | <col_take_point> |        |
```

With the third query, we retrieve and aggregate (generate) string representations for collocations:

```
SELECT DISTINCT ?collocation ?string
WHERE {
  { SELECT ?collocation
    (GROUP_CONCAT(?wrep; separator=" ")
     AS ?string)
    WHERE {
      { SELECT ?collocation ?member
           ?wrep ?order
      WHERE {
        ?collocation a frac:Collocation ;
          ?prop ?member .
        FILTER(?prop=rdfs:member ||
          regex(str(?prop),".*#_[0-9]+$"))
        ?member ((^ontolex:sense)?/
                    ontolex:canonicalForm)?/
                    ontolex:writtenRep ?wrep.
        OPTIONAL {
          ?collocation ?nrel ?member .
          FILTER(regex(str(?nrel),".*#_[0-9]+$"))
          BIND(replace(str(?nrel),".*#_([0-9]+)$",
            "$1")
          AS ?order)
        }
      } GROUP BY ?collocation ?member ?wrep ?order
        ORDER BY ?collocation ?order ?member
      }
    } GROUP BY ?collocation
  }
}
```

The challenge in this query is that the ordering information retrieved above is to be used in an aggregation (in the embedded SELECT statement) by means of GROUP_CONCAT:

```
| collocation      | string       |
===================================
| <col_have_point> | "have point" |
| <col_take_point> | "take point" |
| <col_see_point>  | "see point"  |
```

These surface strings are, indeed, not literally identical to contextualized versions of the corresponding collocations, but they are true to the lexical data in that they implement the VERB + POINT pattern specified in the original dictionary.

5.2. Information Integration for Downstream Applications

Collocations have been used successfully in information integration for downstream applications. One application of collocation is in creating recommendation systems.

To enhance the user experience when using e-commerce platforms, in (Wang and Qiu, 2021) the authors propose a novel fashion collocation recommendation model. The solution uses textual descriptions, purchase data, and category information of items to 1) build a a knowledge graph for modeling the purchase data and category information of items, 2) create knowledge embeddings from the graph, and 3) design a fashion collocation recommendation model that computes the probability of fashion collocation between items to recommend to users. In (Mao et al., 2018), an expert system is designed for costume recommendations which provides customers clothing collocation as recommendations. The system inference engine employs designed rules and user related facts (i.e., physical characteristics) to match customers preferences and generates a clothing recommendation list. @Collocation are also used in recommending news articles to users.

In (Kompan and Bieliková, 2011), the authors include collocations into the preprocessing steps used in text mining to create a fast news articles recommendation system. The system relies on collocations extracted from the articles' characteristics, e.g., title, content, topics, etc., to recommend news content to users.

In (Chu and Wang, 2018), the authors build a collocation corpus for academic writing in engineering and science fields which is used for establishing a sentence-wide collocation recommendation and error detection system for academic writing. After extracting the collocations from sentences, they are classified to create the collocation corpus. The corpus is then used to create a recommendation system for collocations that is also able to detect collocation errors at sentence level.

6. Summary and Discussion

With the collocation extensions for OntoLex-FrAC introduced in this paper, we provide an RDF vocabulary for collocation dictionaries and automated methods of collocation analysis, established as a consensus over contributions from five different institutions and numerous data sets, with the goal of eliciting feedback from reviewers, workshop audience and the scientific community in preparation of the final consolidation of the OntoLex-FrAC module, publication of which as a W3C community report is foreseen for the end of this year.

The key benefit of modelling lexical resources in OntoLex is two-fold:

- It allows us to provide data in a form that can be easily re-used by clients and applications. They can be processed by standard RDF tools and Linguistic Linked Open Data (LLOD) technology. This includes the application of SPARQL for querying distributed lexical data sets.

- It allows to integrate and link such data from distributed and remote sources on the web. Again, this functionality is also integrated in SPARQL (with keywords such as `SERVICE`, `FROM`, or `LOAD`).

With the collocation vocabulary of OntoLex-FrAC, an important contribution has been made in that, now, machine-readable (editions of) traditional collocation dictionaries and collocation scores (automatically generated, either on a fly by a web service, or, as illustrated here, from an existing web portal) can be modelled in the same vocabulary, and can be seamlessly integrated with each other. In comparison to the current capabilities of both TEI (addressing the requirements for collocation *dictionaries* as emerging from traditional lexicographic research) and collocation scores (as generated by tools like SketchEngine or provided by portals such as the Leipzig Wortschatz portal), OntoLex-FrAC covers *both* the needs of developers and APIs (collocation scores, lacking in TEI) and the needs of the lexicographer (modelling dictionaries and their lexicographic structure by means of OntoLex and OntoLex-Lexicog – lacking in Wortschatz or SketchEngine).
Although these additions to OntoLex-FrAC appear to be minimal (one new class for collocations, one new object property to identify their head, one new datatype property to represent collocations cores – and its large and extensible set of subproperties), they have been shown to be sufficient and to be sufficiently generic to model both collocation dictionaries and API/collocation score requirements.

Acknowledgments

The research described in this paper was conducted in the context of the Cost Action CA18209 *Nexus Linguarum. European network for Web-centred linguistic data science* and innovation programme via the project Prêt-à-LLOD (2019-2022, grant agreement no. 825182, Maxim Ionov). Moreover, the authors would like to thank Julia Bosque-Gil for contributing to the development of OntoLex-FrAc as well as to all OntoLex-FrAC contributors.

7. Bibliographical References

Bosque-Gil, J. and Gracia, J. (2019). The OntoLex Lemon Lexicography Module. Final Community Group Report. Technical report, W3C.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997). Dynamic Itemset Counting and Implication Rules for Market Basket Data. In Joan Peckham, editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pages 255–264. ACM Press.

Chiarcos, C., Ionov, M., de Does, J., Depuydt, K., Khan, F., Stolk, S., Declerck, T., and McCrae, J. P. (2020). Modelling frequency and attestations for ontolex-lemon. In *Proceedings of the 2020 Globalex Workshop on Linked Lexicography*, pages 1–9.

Chiarcos, C., Declerck, T., and Ionov, M. (2021). Embeddings for the Lexicon: Modelling and Representation. In *Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6)*, pages 13–19.

Chu, Y.-L. and Wang, T.-I. (2018). A Sentence-Wide Collocation Recommendation System with Error Detection for Academic Writing. In *Lecture Notes in Computer Science*, pages 307–316. Springer International Publishing.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765.

Hardie, A. (2012). CQPweb. Combining power, flexibility and usability in a corpus analysis tool. *International journal of corpus linguistics*, 17(3):380–409.

Initiative, T. E. (2022). P5: Guidelines for electronic text encoding and interchange, chap. 9 dictionaries. Technical report. Version 4.4.0. Last updated on 19th April 2022, revision ff9cc28b0.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlỳ, P., and Suchomel, V. (2014). The Sketch Engine: ten years on. *Lexicography*, 1(1):7–36.

Kompan, M. and Bieliková, M. (2011). News Article Classification Based on a Vector Representation Including Words' Collocations. In *Advances in Intelligent and Soft Computing*, pages 1–8. Springer Berlin Heidelberg.

Larose, D. T. and Larose, C. D., (2014). *Association Rules*, chapter 12, pages 247–265. John Wiley and Sons, Ltd.

Manning, C. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press Ltd, May.

Mao, Q., Dong, A., Miao, Q., and Pan, L. (2018). Intelligent Costume Recommendation System Based on Expert System. *Journal of Shanghai Jiaotong University (Science)*, 23(2):227–234, apr.

McCrae, J. P., Bosque-Gil, J., Gracia, J., Buitelaar, P., and Cimiano, P. (2017). The Ontolex-Lemon model: development and applications. In *Proceedings of eLex 2017 conference*, pages 19–21.

OUP. (2002). *Oxford Collocations Dictionary for Students of English*. Oxford University Press, USA.

Pedersen, T. (1998). Dependent bigram identification. *AAAI/IAAI*, 1197.

Role, F. and Nadif, M. (2011). Handling The Impact of Low Frequency Events on Co-Occurrence Based Measures of Word Similarity - A Case Study of Pointwise Mutual Information. In *Proceedings of* *the International Conference on Knowledge Discovery and Information Retrieval - Volume 1: KDIR, (IC3K 2011)*, pages 218–223. INSTICC, SciTePress.

Suchánek, M. and Pergl, R. (2020). Case-study-based review of approaches for transforming UML class diagrams to OWL and vice versa. In *2020 IEEE 22nd Conference on Business Informatics (CBI)*, volume 1, pages 270–279. IEEE.

Wang, S. and Qiu, J. (2021). A deep neural network model for fashion collocation recommendation using side information in e-commerce. *Applied Soft Computing*, 110:107753, oct.