UniDive WG2 / T2.3 Standardizing Lexica of MWEs

Christian Chiarcos

christian.chiarcos@uni-a.de

Applied Computational Linguistics (ACoLi)

University of Augsburg, Germany

UniDive MoU: WG2 Lexicon-Corpus Interface

- **T2.1** Cross-language unification of lexical features: (i) harmonizing the definition of a “syntactic word” across languages, (ii) harmonizing lemmatization rules (for words and MWEs) and lexical features across languages, (iii) standardizing lists of lexemes for auxiliaries, pronouns and determiners;
- **T2.2** Design of a lexicon-corpus interface aiming at: (i) interlinking MWE lexicon entries with their occurrences in corpora, (ii) cross-lingually unified lexicography of idiosyncratic constructions;
- **T2.3** Proof-of-concept lexical encoding of MWEs following the above design.

UniDive MoU: WG2 Lexicon-Corpus Interface

- **T2.1** Cross-language unification of lexical features: (i) harmonizing the definition of a “syntactic word” across languages, (ii) harmonizing lemmatization rules (for words and MWEs) and lexical features across languages, (iii) standardizing lists of lexemes for auxiliaries, pronouns and determiners; **=> focus on linguistics**
- **T2.2** Design of a lexicon-corpus interface aiming at: (i) interlinking MWE lexicon entries with their occurrences in corpora, (ii) cross-lingually unified lexicography of idiosyncratic constructions; **=> focus on modelling**

T2.3 Proof-of-concept lexical encoding of MWEs following the above design.

=> focus on data and technologies

UniDive MoU: WG2, T2.3

- **T2.1** Cross-language unification of lexical features: (i) harmonizing the definition of a “syntactic word” across languages, (ii) harmonizing lemmatization rules (for words and MWEs) and lexical features across languages, (iii) standardizing lists of lexemes for auxiliaries, pronouns and determiners; **=> focus on linguistics**
- **T2.2** Design of a lexicon-corpus interface aiming at: (i) interlinking MWE lexicon entries with their occurrences in corpora, (ii) cross-lingually unified lexicography of idiosyncratic constructions; **=> focus on modelling**

T2.3 Proof-of-concept lexical encoding of MWEs following the above design.

=> focus on data and technologies

- At the first general meeting, we decided that T2.3 will start by looking into two facets
  - survey of existing MWE resources and their requirements/design
  - explore existing standards

UniDive MoU: WG2, T2.3

- **T2.1** Cross-language unification of lexical features: (i) harmonizing the definition of a “syntactic word” across languages, (ii) harmonizing lemmatization rules (for words and MWEs) and lexical features across languages, (iii) standardizing lists of lexemes for auxiliaries, pronouns and determiners; **=> focus on linguistics**
- **T2.2** Design of a lexicon-corpus interface aiming at: (i) interlinking MWE lexicon entries with their occurrences in corpora, (ii) cross-lingually unified lexicography of idiosyncratic constructions; **=> focus on modelling**

T2.3 Proof-of-concept lexical encoding of MWEs following the above design.

=> focus on data and technologies

- At the first general meeting, we decided that T2.3 will start by looking into two facets
  - survey of existing MWE resources and their requirements/design **[=> Stella & Ivelina]**
  - explore existing standards **[=> now: high-level overview]**

Selected features of standards for (MWE) dictionaries

- user and developer friendliness
  - How easy to read and write (without specialized software)?
  - How easy to produce, store and process in downstream applications?
- expressivity
  - Can we encode what we need without idiosyncratic/ad hoc extensions?
- normativity
  - Do/can we use a controlled vocabulary?
- genericity / coverage
  - Is there a systematic way to extend the vocabulary for unexpected cases?
- linkability
  - Can we link external data (corpora, other dictionaries, ...) without building specialized software?

Three main families of standards / conventions

- **focus on tables** (TSV, CSV, SQL/RDBMS)
  - usage: dominant database paradigm
  - examples:* PanLex, CLLD, Global WordNet

* none of these are specifically tailored towards MWEs.
This is true for all examples listed in the following.

Three main families of standards / conventions

- **focus on tables** (TSV, CSV, SQL/RDBMS)
  - usage: dominant database paradigm
- **focus on documents** (XML => JSON)
  - usage (XML): widely used in lexicography, DH, and for language resources
  - usage (JSON): API development, modern document stores
  - examples: TEI (-Dict; Lex-0), LMF, TBX, XDXF

Three main families of standards / conventions

- **focus on tables** (TSV, CSV, SQL/RDBMS)
  - usage: dominant database paradigm
- **focus on documents** (XML => JSON)
  - usage (XML): widely used in lexicography, DH, and for language resources
- **focus on information integration** (GraphDBs, Linked Data)
  - usage
    - used for interlinking lexical data sets with each other, with knowledge graphs and other external data (examples: DBnary, Wikidata)
    - wrapper technologies for other kinds of data (example: CLLD)
  - schema-free: needs to be complemented by a controlled vocabulary
    - for lexical data, this is primarily OntoLex-Lemon
  - examples: 3 posters yesterday ;)

Three main families of standards / conventions

- **focus on tables** (TSV, CSV, SQL/RDBMS)
  - usage: dominant database paradigm
- **focus on documents** (XML => JSON)
  - usage (XML): widely used in lexicography, DH, and for language resources
- **focus on information integration** (GraphDBs, Linked Data)
  - usage
    - used for interlinking lexical data sets with each other, with knowledge graphs and other external data (examples: DBnary, Wikidata)

- emerging standards with multiple serializations
  - DMLex/Lexidma (https://github.com/oasis-tcs/lexidma/releases/download/dev-latest/dmlex-v1.0-wd01.pdf)

User and Developer Friendliness

- user and developer friendliness
  - How easy to read and write (without specialized software)?
  - How easy to produce, store and process in downstream applications?
- expressivity
  - Can we encode what we need without idiosyncratic/ad hoc extensions?
- normativity
  - Do/can we use a controlled vocabulary?
- genericity / coverage
  - Is there a systematic way to extend the vocabulary for unexpected cases?
- linkability
  - Can we link external data (corpora, other dictionaries, ...) without building specialized software?

Tabular Formats: UniMorph

- Source: https://unimorph.github.io/
- **Single-table format**
- some MWE support for analytic inflection

```
afsteken	afsteken	V;NFIN
afsteken	steek af	V;IND;SG;1;PRS
afsteken	steek af	V;IND;SG;1;PST
afsteken	afsteek V;IND;SG;1;PRS;LGSPEC02
afsteken	afsteek V;IND;SG;1;PST;LGSPEC02
afsteken	stickst af	V;IND;SG;2;PRS
afsteken	steekst af	V;IND;SG;2;PST
afsteken	afstickst	V;IND;SG;2;PRS;LGSPEC02
afsteken	afsteekst	V;IND;SG;2;PST;LGSPEC02
afsteken	stickt af	V;IND;SG;3;PRS
afsteken	steek af	V;IND;SG;3;PST
afsteken	afstickt	V;IND;SG;3;PRS;LGSPEC02
afsteken	afsteek V;IND;SG;3;PST;LGSPEC02
afsteken	steekt af	V;IND;PL;PRS
afsteken	steken af	V;IND;PL;PST
afsteken	afsteekt	V;IND;PL;PRS;LGSPEC02
afsteken	afsteken	V;IND;PL;PST;LGSPEC02
afsteken	steek af	V;IMP;SG;PRS
afsteken	steekt af	V;IMP;PL;PRS
afsteken	afsteken	V.PTCP;PRS
```

Tabular Formats: UniMorph

- **user friendliness**: +++
- **developer friendliness**: +++
- **expressivity**: -
- **normativity**: (+)
- **genericity/extensibility**: (-)

  [but we can extend to a multi-table format]
- **linkability**: (-)

  [requires coding or wrapper technologies]

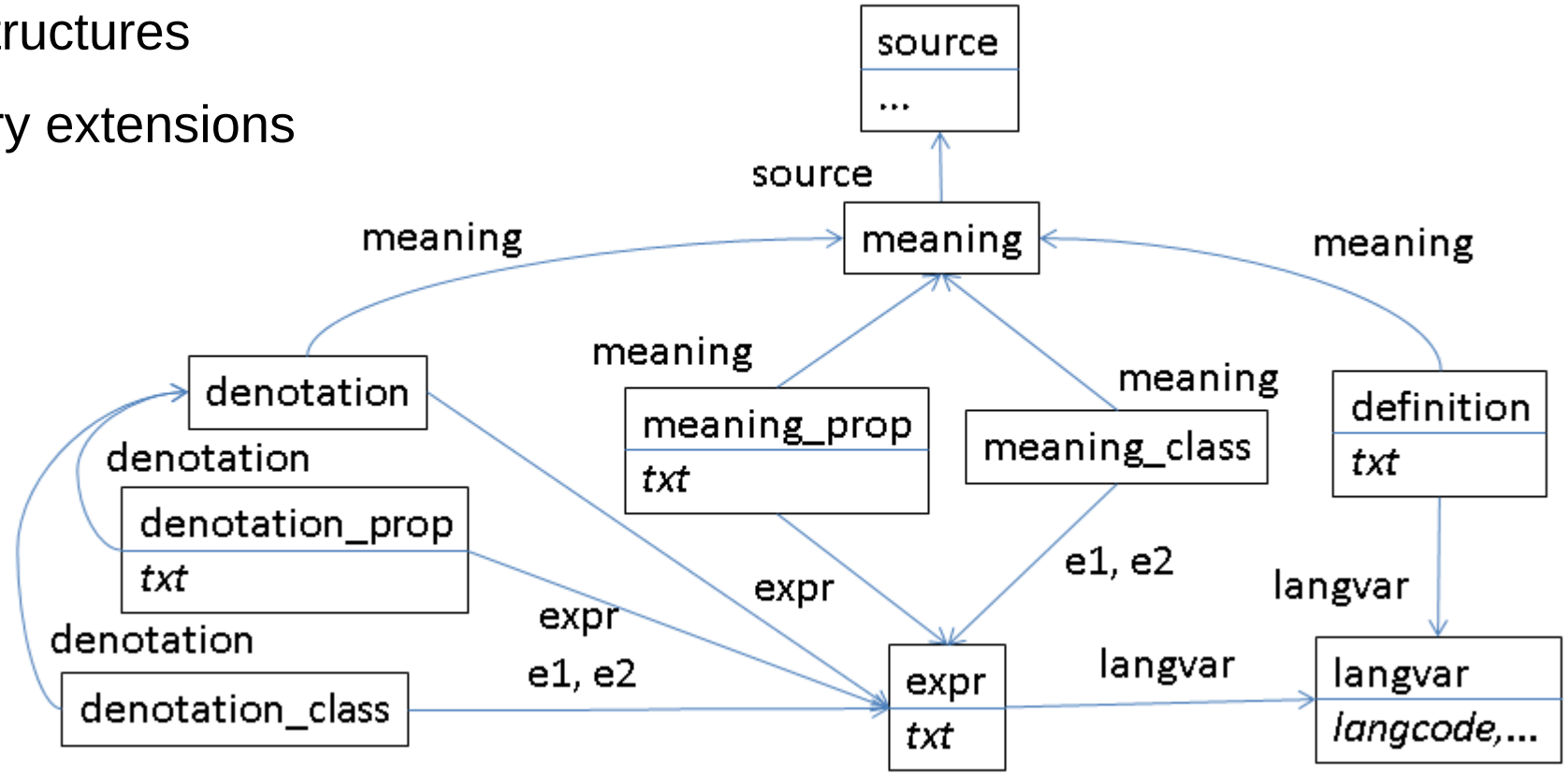
```
afsteken    afsteken        V;NFIN
afsteken    steek af        V;IND;SG;1;PRS
afsteken    steek af        V;IND;SG;1;PST
afsteken    afsteek V;IND;SG;1;PRS;LGSPEC02
afsteken    afsteek V;IND;SG;1;PST;LGSPEC02
afsteken    stickst af      V;IND;SG;2;PRS
afsteken    steekst af      V;IND;SG;2;PST
afsteken    afstickst       V;IND;SG;2;PRS;LGSPEC02
afsteken    afsteekst       V;IND;SG;2;PST;LGSPEC02
afsteken    stickt af       V;IND;SG;3;PRS
afsteken    steek af        V;IND;SG;3;PST
afsteken    afstickt        V;IND;SG;3;PRS;LGSPEC02
afsteken    afsteek V;IND;SG;3;PST;LGSPEC02
afsteken    steekt af       V;IND;PL;PRS
afsteken    steken af       V;IND;PL;PST
afsteken    afsteekt        V;IND;PL;PRS;LGSPEC02
afsteken    afsteken        V;IND;PL;PST;LGSPEC02
```

As long as a problem is simple enough to encode it in a simple table, this is preferred

... I am not sure that MWEs are, though ...

Tabular Formats: PanLex

- Source: https://panlex.org/
- **RDBMS dump** (CSV or JSON)
- 2500 dictionaries
- table regulates core data structures
- special tables for vocabulary extensions



Tabular Formats: PanLex

- **user friendliness**: (-) [you don’t want to write this by hand]
- **developer friendliness**: +++ [that’s the default approach]
- **expressivity**: + [special tables for vocabulary extensions]
- **normativity**: (+) [only for core data structures]
- **genericity/extensibility**: (+) [extensible, but not with a controlled vocabulary]
- **linkability**: (-) [requires coding or wrapper technologies]

Very capable backend technology, but requires some investment into developing special-purpose software.

Document-Centered Formats: FreeDict

- Source: https://freedict.org/
- 140 dictionaries
- Vocabulary: XML/TEI (TEI-Dict)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="freedict-dictionary.css"?>
<?oxygen RNGSchema="freedict-P5.rng" type="xml"?>
<!DOCTYPE TEI SYSTEM "freedict-P5.dtd">
<TEI xmlns="http://www.tei-c.org/ns/1.0" xmlns:wikdict="http://www.wikdict.com/ns/1.0">
 <teiHeader xml:lang="en">
 <text>
  <body xml:lang="it">
   <entry>
   <form>
    <orth>crazia</orth>
    <pron>(/'krattsja/</pron>
   </form>
   <gramGrp>
    <pos>suffix</pos>
   </gramGrp>
   <sense>
   <cit type="trans" xml:lang="bg">
    <quote>кра́ция</quote>
   </cit>
   <sense>
    <def>Vedi le traduzioni</def>
   </sense>
  </sense>
 </entry>
```

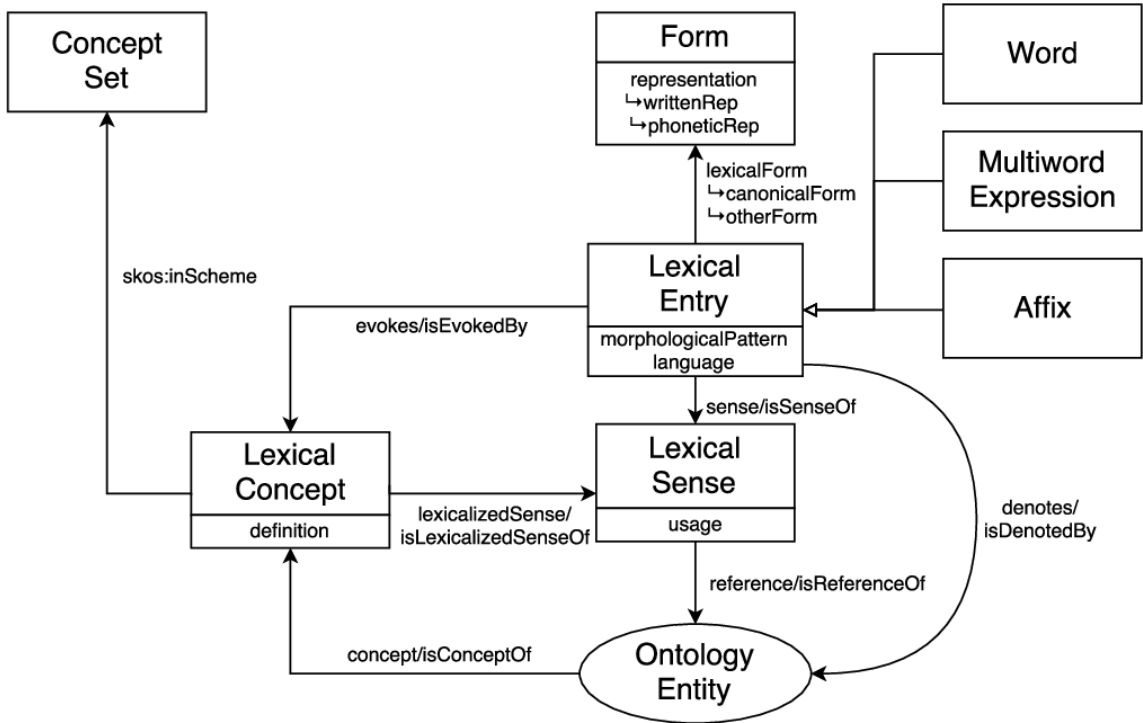
Document-Centered Formats: FreeDict

- **user friendliness**: + [you get used to it fairly quicky]
- **developer friendliness**: + [still sufficiently supported, but some entry barrier]
- **expressivity**: + [if necessary, you can resort to *entryFree*]
- **normativity**: + [RNG Schema]
- **genericity/extensibility**: - [not extensible, unless the schema is extended]
- **linkability**: (-) [custom TEI Pointer structures, not supported by off-the-shelf technology]

If you have an adequate schema, you can also encode complex information in a relatively user-friendly way. Editing doesn't require specialized software.

Linked Data: ACoLi Dictionary Graph

- Source: https://github.com/acoli-repo/acoli-dicts
- >3000 bilingual dictionaries
- Vocabulary: OntoLex
- Custom export to tabular data with SPARQL
- most data is not natively created in this format, but converted / wrapped from tabular formats, XML or semistructured data



Linked Data: OntoLex

- **user friendliness**: - [you *really* don’t want to write or read this directly]
- **developer friendliness**: + [fairly well supported, but has some entry barrier]
- **expressivity**: + [add your own information in your own namespace]
- **normativity**: + [OntoLex vocabulary, SHACL validation]
- **genericity/extensibility**: + [add your own information in your own namespace]
- **linkability**: +++ [primary purpose of this technology, also extends to data in other formats, if an RDF wrapper is provided]

This is backend technology for sharing, linking and querying data.

Using SPARQL, we can query directly or flexibly generate tables for further processing

Using standardized wrapper technologies, we can access and process other formats.

Interim Summary

	single table (UniMorph)	multi-table (PanLex)	XML (~ JSON) (FreeDict)	RDF Graphs (OntoLex)
user friendliness	+++	-	+	-
developer friendliness	+++	+++	+	+
expressivity	-	+	+	+
normativity	(+)	(+)	+	+
genericity/extensibility	(-)	(+)	-	+
linkability	(-)	(-)	(-)	+++

Note that this is just my personal assessment. Feel free to disagree ;)

As it stands, I don't see* **one solution** for both

1) creating and maintaining MWE dictionaries, and

2) linking MWE dictionaries (with each other, corpora, KGs)

* At least not without significant investments in software development.

Within COST, it seems more realistic to keep these aspects apart and to map/wrap/convert

Current Considerations

- Yesterday, we discussed a new division of labor within T2.3
  - or, alternatively, a separation between T2.3 and a novel task specifically on linking
- This doesn’t mean to abandon ties, but it means to more clearly distinguish two different strands of current activities
  - surveying and consolidating existing data (see next session)
  - converting/wrapping and linking that data (in the novel [sub-]task)
    - we have a number of encouraging experiments in this direction (e.g., yesterday’s posters by Ranka et al. and me et al.; Ranka and me could be coordinating this)
- These continue to be interrelated with each other and with the other WG2 tasks.
- Any expressions of interest? Other feedback? Questions?