Gilles Sérasset & Andon Tchechmedjiev

The DBnary ecosystem

Data, tools and APIs



The DBnary Dataset

DBnary is to Wiktionary
What
DBpedia is to Wikipedia

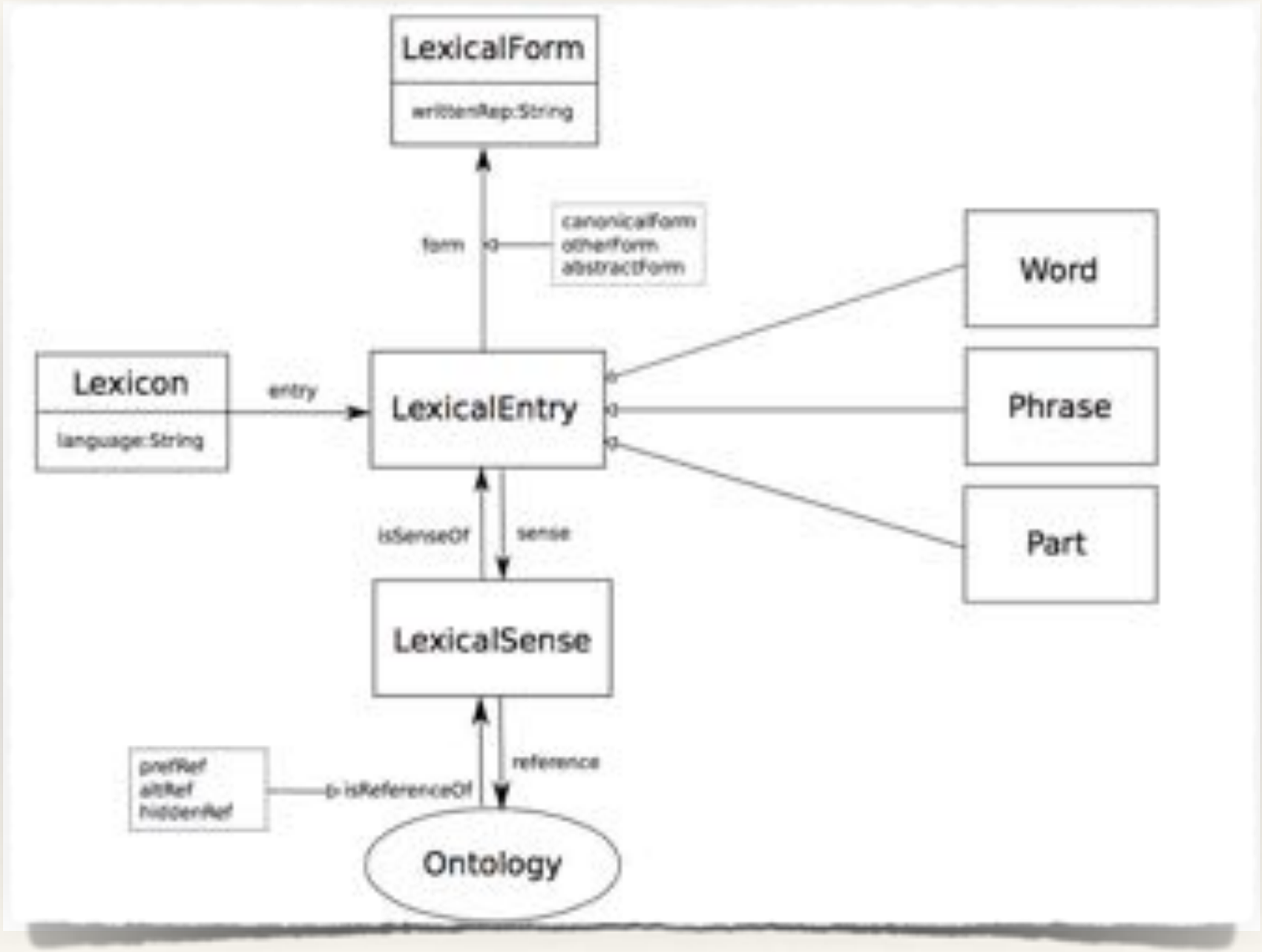
The DBnary Dataset

- Lexical data extracted from 18 20 21 different Wiktionary language editions
  - → 2 more in alpha version.
- Extracted data is available as Linked Data
  - → uses: `lemon` (+ few dbnary ext.), `lexinfo`, `lexvo` and `OLiA`.
  - → Coming Soon: support for latest `ontolex` spec.

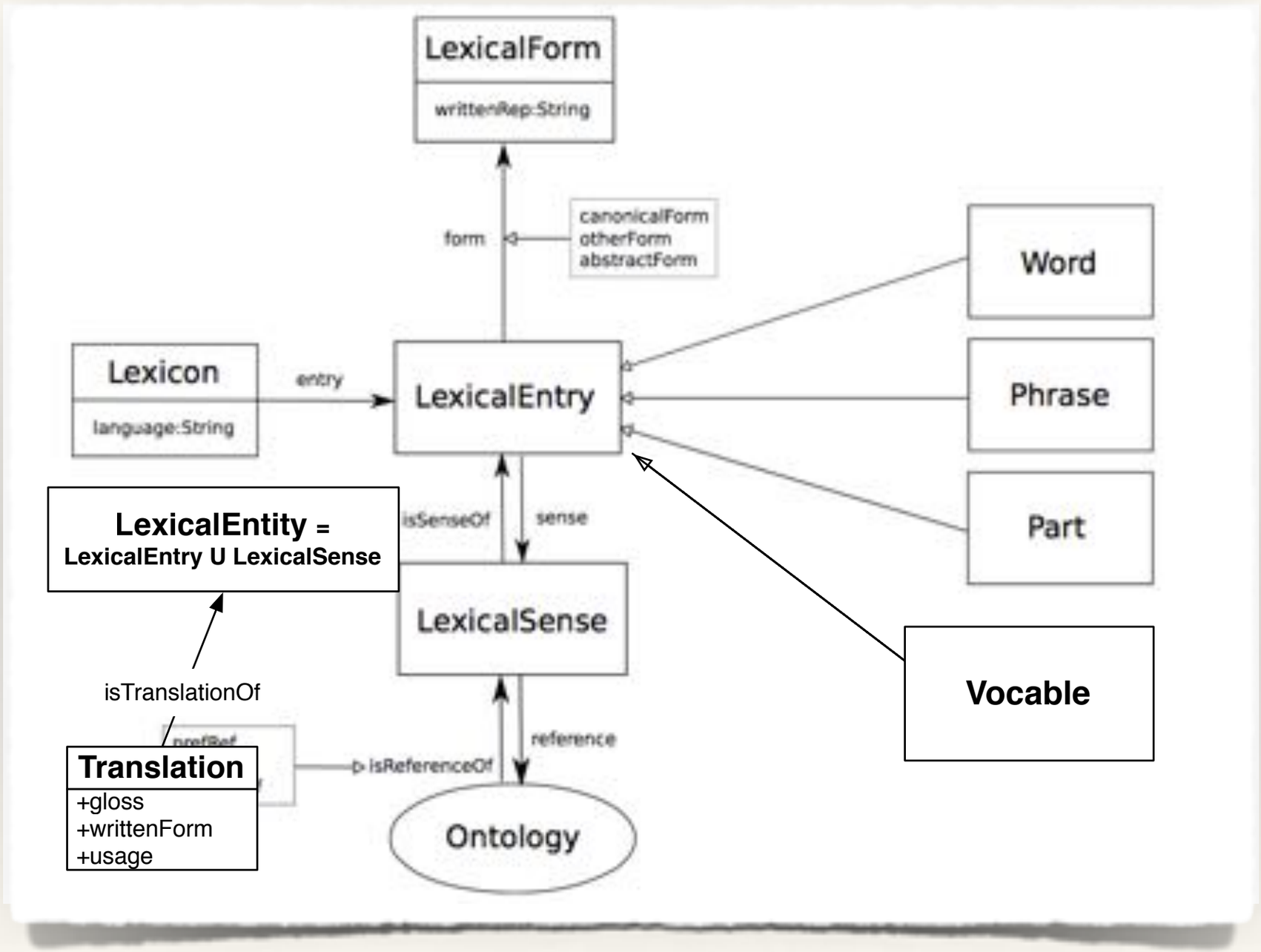
The DBnary Dataset

- **Core data**
  - Lexical Entries, Lexical Senses and Translations
- **Additional data**
  - Semantically enriched Relations
    - Translations are attached to their source Lexical Sense when possible
    - Lexico-semantic relations are also attached to their source Lexical Sense
  - Morphology
    - Extensive representation of morphology (a set of “lemon:otherForm”)

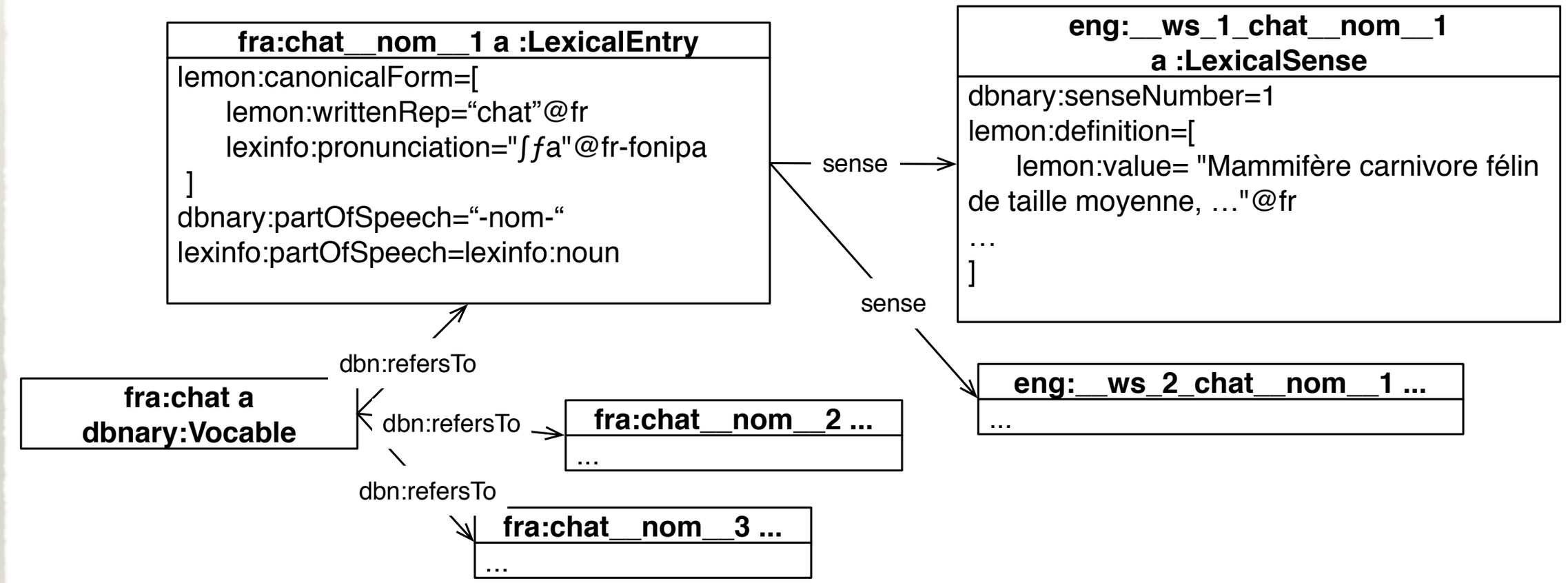
The DBnary Dataset



The DBnary Dataset



Example: core data



Example: core data

Vocables

```
fra:chat   a                  dbnary:Vocable ;
           dbnary:refersTo    fra:chat__nom__1 , fra:chat__nom__2 … .

deu:Katze  a                  dbnary:Vocable ;
           dbnary:refersTo    deu:Katze__Substantiv__1 .
```

Example: core data

Lexical Entries

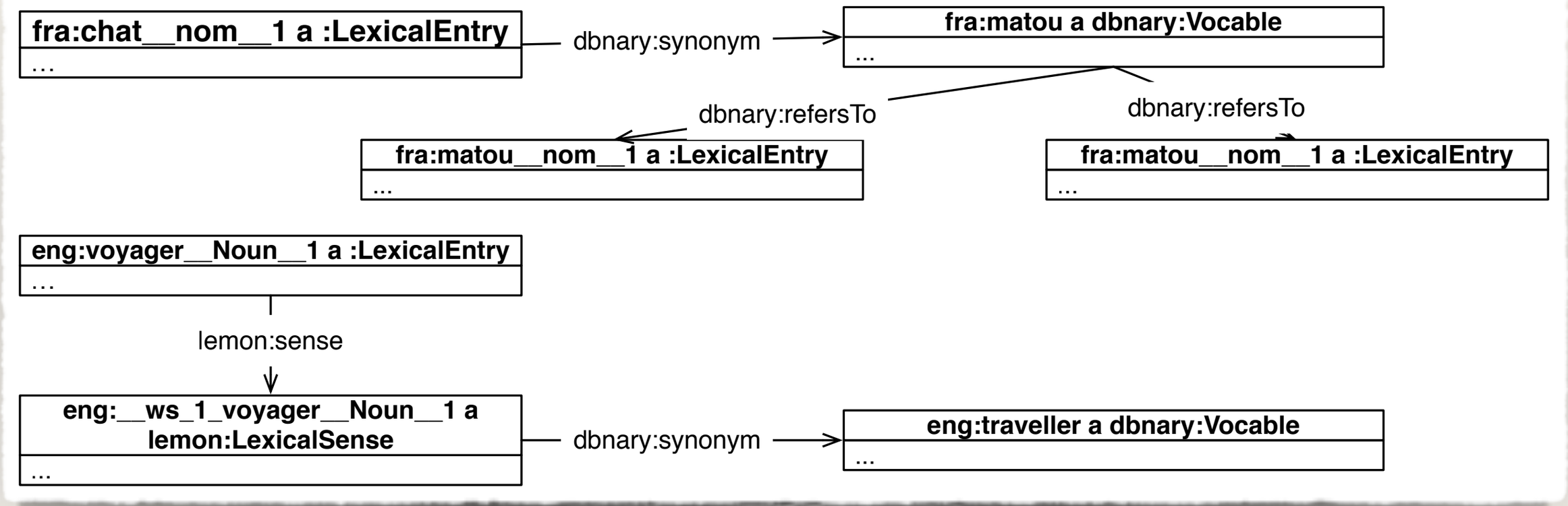
```
fra:chat__nom__1  a                  lemon:LexicalEntry , lemon:Word ;
        dbnary:hypernym              fra:félidé ;
        dbnary:hyponym               fra:chat_des_sables , fra:chat_à_pieds_noirs , … ;
        dbnary:partOfSpeech          "-nom-" ;
        dbnary:synonym               fra:Raminagrobis , fra:minet , fra:matou … ;
        dcterms:language             lexvo:fra ;
        lemon:canonicalForm          [ lemon:writtenRep        "chat"@fr ;
                                       lexinfo:number          lexinfo:singular ;
                                       lexinfo:pronunciation   "ʃa"@fr-fonipa
                                     ] ;
        lemon:language               "fr" ;
        lemon:lexicalVariant         [ lemon:writtenRep  "béarnais : gat"@fr ] ;
        lemon:sense                  fra:__ws_8_chat__nom__1 , fra:__ws_4_chat__nom__1 , … ;
        lexinfo:partOfSpeech         lexinfo:noun .
```

Example: core data

Lexical Senses

```
fra:__ws_1_chat__nom__1
        a                       lemon:LexicalSense ;
        dbnary:senseNumber      "1"^^<http://www.w3.org/2001/XMLSchema#string> ;
        lemon:definition        [ lemon:value  "Mammifère carnivore félin de taille moyenne, …"@fr ] ;
        lemon:example           [ dbnary:exampleSource  "H. G. Wells, La Guerre dans les airs, 1908,
traduit par Henry-D. Davray & B. Kozakiewicz, Mercure de France, 1921, page 335"@fr ;
                                  lemon:value           "Soudain, d’un seul élan, cela se précipita sur
lui, avec un miaulement plaintif et la queue droite. C’était un jeune chat, menu et décharné, qui
frottait sa tête contre les jambes de Bert, en ronronnant."@fr
                                ] ;
        lemon:example           [ dbnary:exampleSource  "Octave Mirbeau, Contes cruels : Mon
oncle"@fr ;
                                  lemon:value           "Entre autres manies, mon oncle avait celle de
tuer tous les chats qu’il rencontrait. Il faisait, à ces pauvres bêtes, une chasse impitoyable, une
guerre acharnée de trappeur."@fr
                                ] ;
        lemon:example           [ dbnary:exampleSource  "Erckmann-Chatrian, Histoire d’un conscrit de
1813, J. Hetzel, 1864"@fr ;
                                  lemon:value           "[…] le chat gris, un peu sauvage, nous
regardait de loin, à travers la balustrade de l’escalier au fond, sans oser descendre."@fr
                                ] .
```

Example: core data



Example: core data

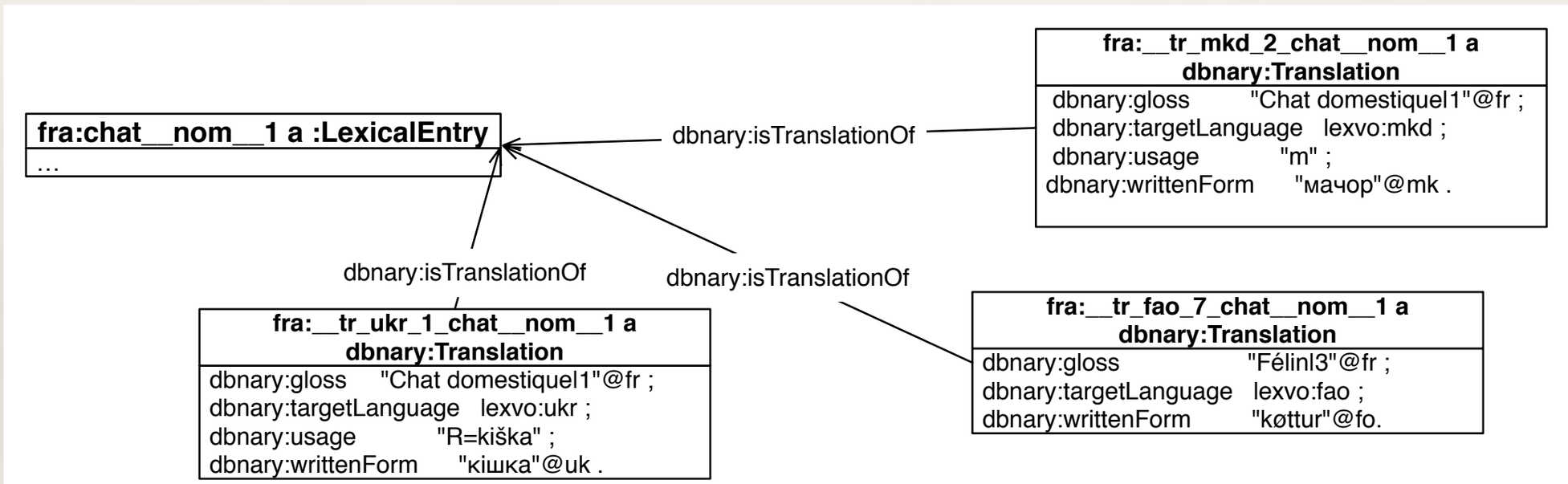
Lexico-semantic relations

```
fra:chat__nom__1  a                  lemon:LexicalEntry , lemon:Word ;
        dbnary:hypernym              fra:félidé ;
        dbnary:hyponym               fra:chat_des_sables , fra:chat_à_pieds_noirs , … ;
        dbnary:synonym               fra:Raminagrobis , fra:minet , fra:matou … ;
        dcterms:language             lexvo:fra ;
        …
        lexinfo:partOfSpeech  lexinfo:noun .

deu:__hypo_45_Katze__Substantiv__1
        a                    rdf:Statement ;
        rdf:object           deu:Schwarzfußkatze ;
        rdf:predicate        dbnary:hyponym ;
        rdf:subject          deu:Katze__Substantiv__1 ;
        dbnary:gloss         "3" .

deu:__hyper_11_Tiger__Substantiv__1
        a                    rdf:Statement ;
        rdf:object           deu:Katze ;
        rdf:predicate        dbnary:hypernym ;
        rdf:subject          deu:Tiger__Substantiv__1 ;
        dbnary:gloss         "2" .
```

Example: core data



Example: core data

Translations

```
fra:__tr_pol_6_chat__nom__1
        a                       dbnary:Translation ;
        dbnary:gloss            "Chat mâle|2"@fr ;
        dbnary:isTranslationOf  fra:chat__nom__1 ;
        dbnary:targetLanguage   lexvo:pol ;
        dbnary:writtenForm      "kot"@pl .

fra:__tr_jpn_1_chat__nom__1
        a                       dbnary:Translation ;
        dbnary:gloss            "Chat domestique|1"@fr ;
        dbnary:isTranslationOf  fra:chat__nom__1 ;
        dbnary:targetLanguage   lexvo:jpn ;
        dbnary:usage            "R=neko" ;
        dbnary:writtenForm      "猫"@ja .

fra:__tr_ita_6_chat__nom__1
        a                       dbnary:Translation ;
        dbnary:gloss            "Félin|3"@fr ;
        dbnary:isTranslationOf  fra:chat__nom__1 ;
        dbnary:targetLanguage   lexvo:ita ;
        dbnary:writtenForm      "felina"@it .
```

Example: Additional data

Enhanced translations

```
fra:__tr_pol_6_chat__nom__1
        dbnary:isTranslationOf  fra:__ws_2_chat__nom__1 .

fra:__tr_jpn_1_chat__nom__1
        dbnary:isTranslationOf  fra:__ws_1_chat__nom__1 .

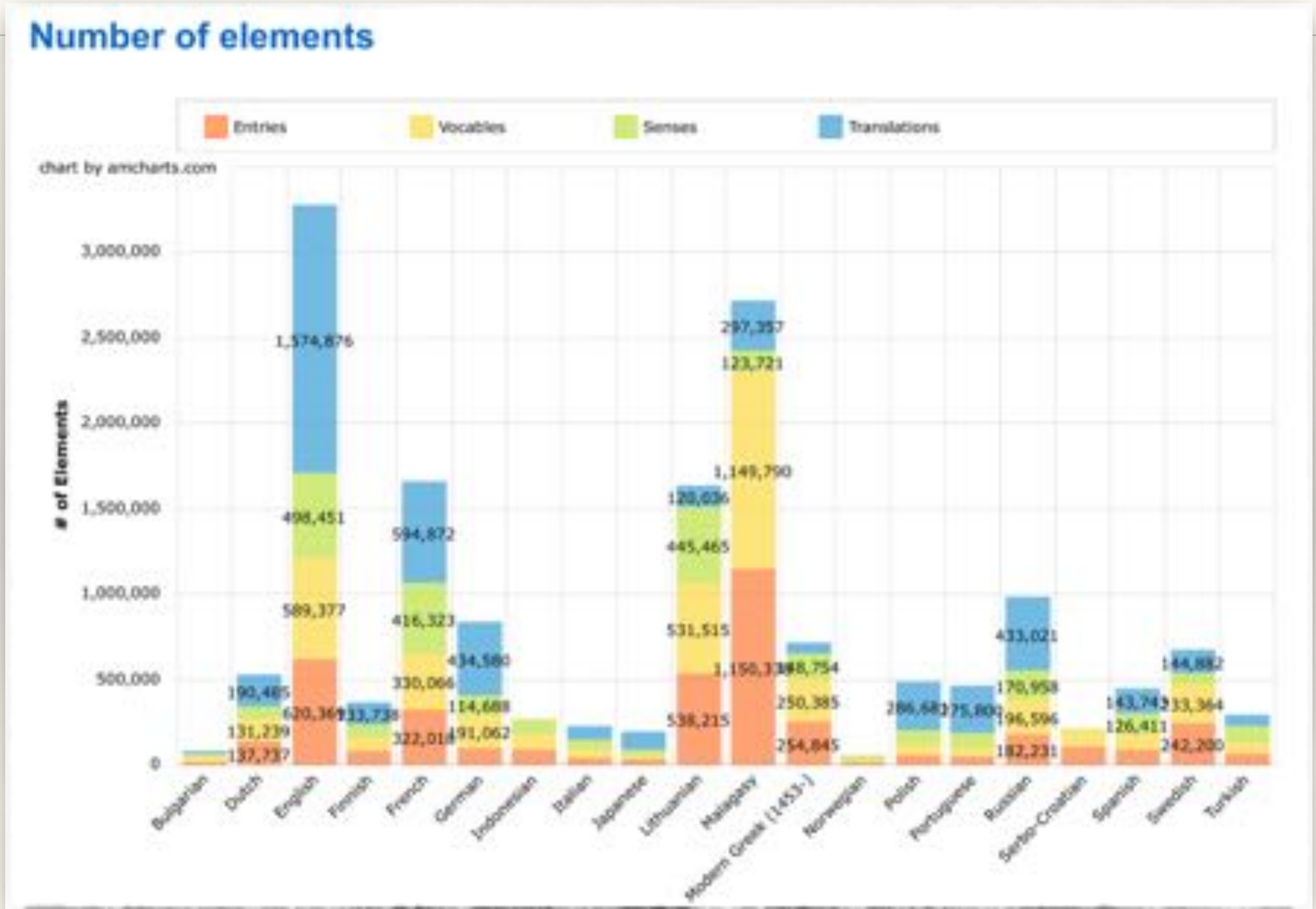
fra:__tr_ita_6_chat__nom__1
        dbnary:isTranslationOf  fra:__ws_3_chat__nom__1 .
```

Example: Additional data

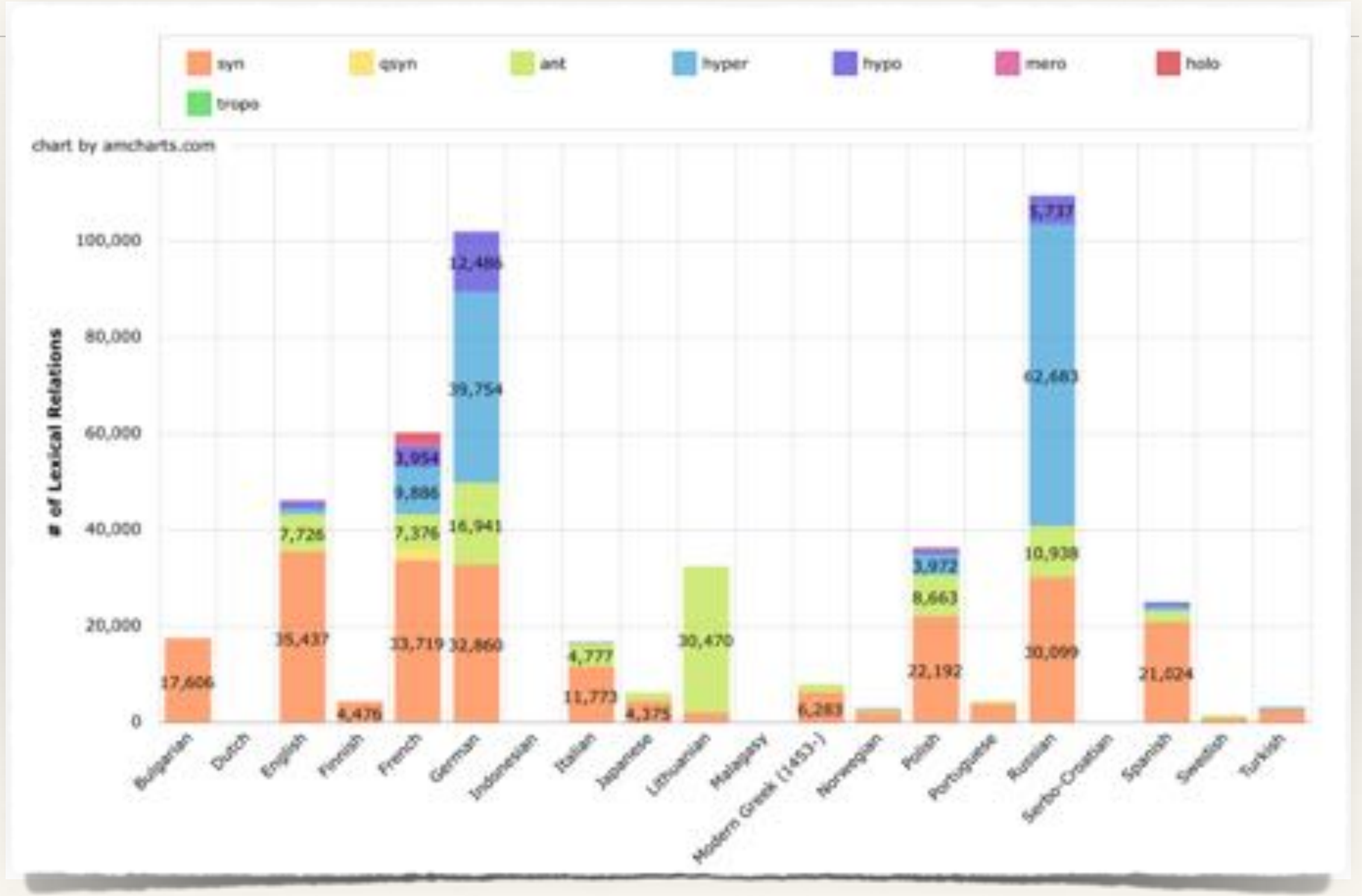
Morphology

```
deu:Katze__Substantiv__1
        lemon:otherForm  [ olia:hasCase      olia:Accusative ;
                           olia:hasNumber    olia:Plural ;
                           lemon:writtenRep  "die Katzen"@de
                         ] ;
        lemon:otherForm  [ olia:hasCase      olia:Accusative ;
                           olia:hasNumber    olia:Singular ;
                           lemon:writtenRep  "die Katze"@de
                         ] ;
        lemon:otherForm  [ olia:hasCase      olia:DativeCase ;
                           olia:hasNumber    olia:Plural ;
                           lemon:writtenRep  "den Katzen"@de
                         ] ;
        lemon:otherForm  [ olia:hasCase      olia:DativeCase ;
                           olia:hasNumber    olia:Singular ;
                           lemon:writtenRep  "der Katze"@de
                         ] ;
        …
        lemon:otherForm  [ olia:hasCase      olia:Nominative ;
                           olia:hasNumber    olia:Singular ;
                           lemon:writtenRep  "die Katze"@de
                         ] ;
```

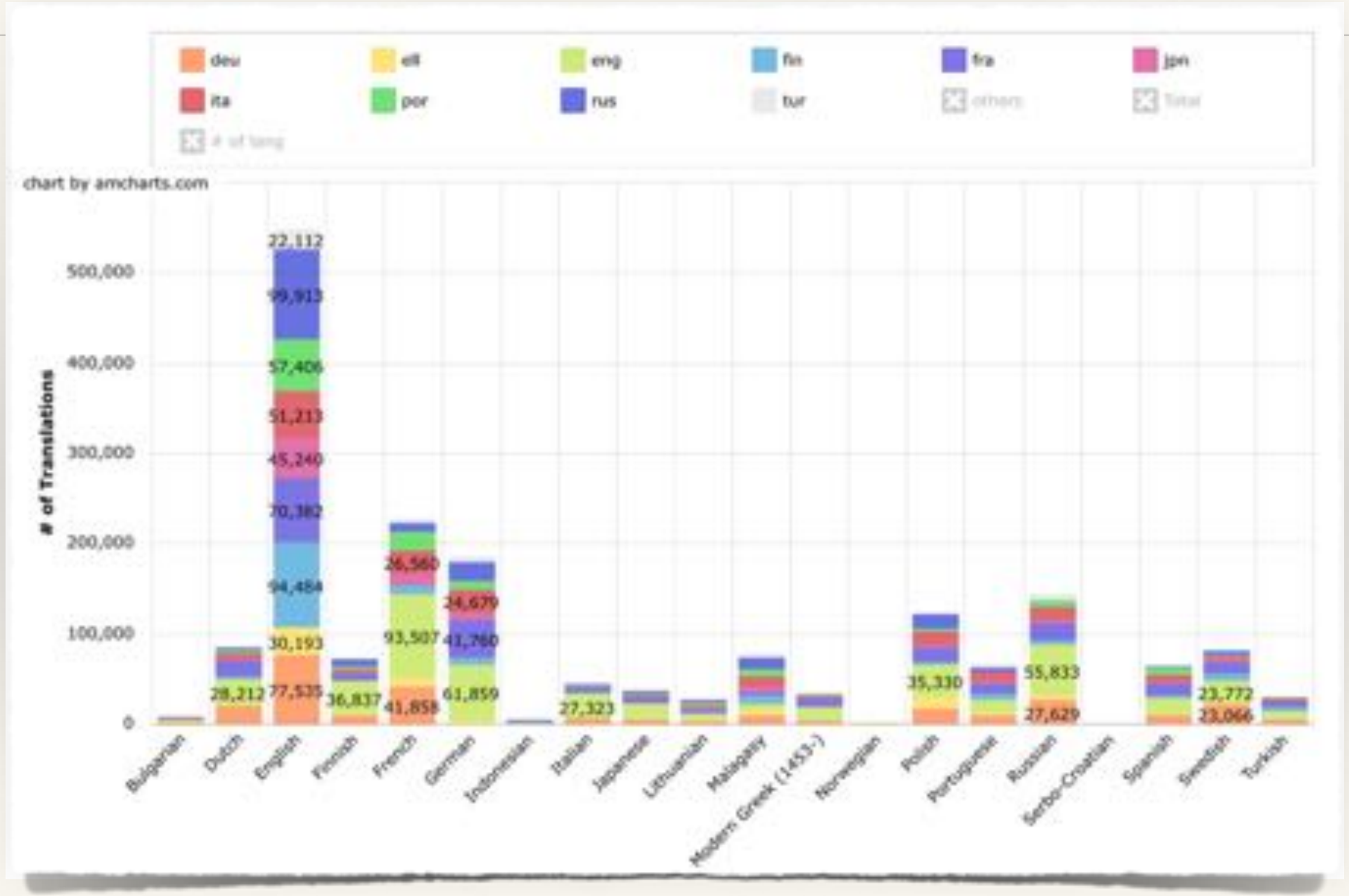
Size of the data



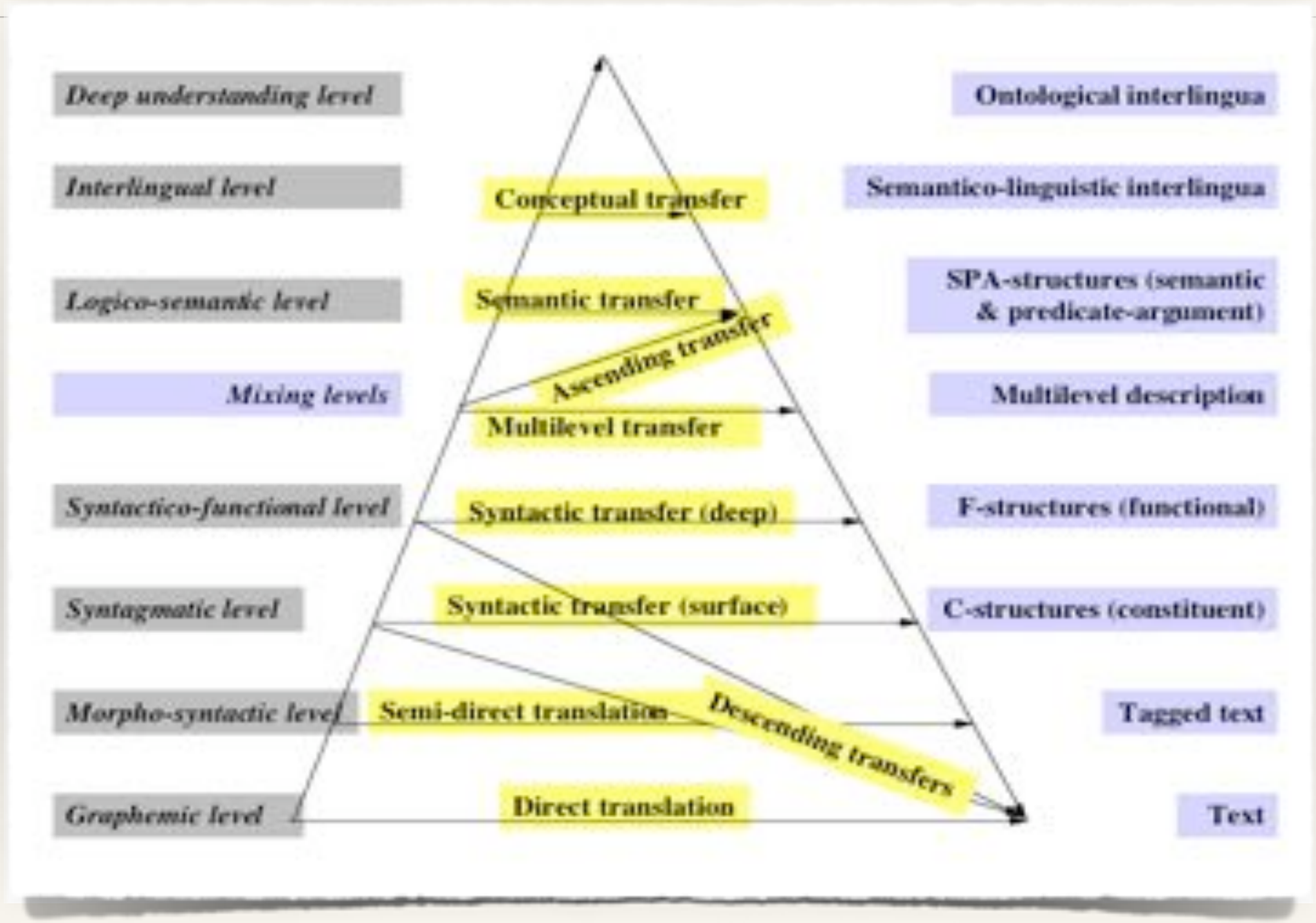
Size of the data

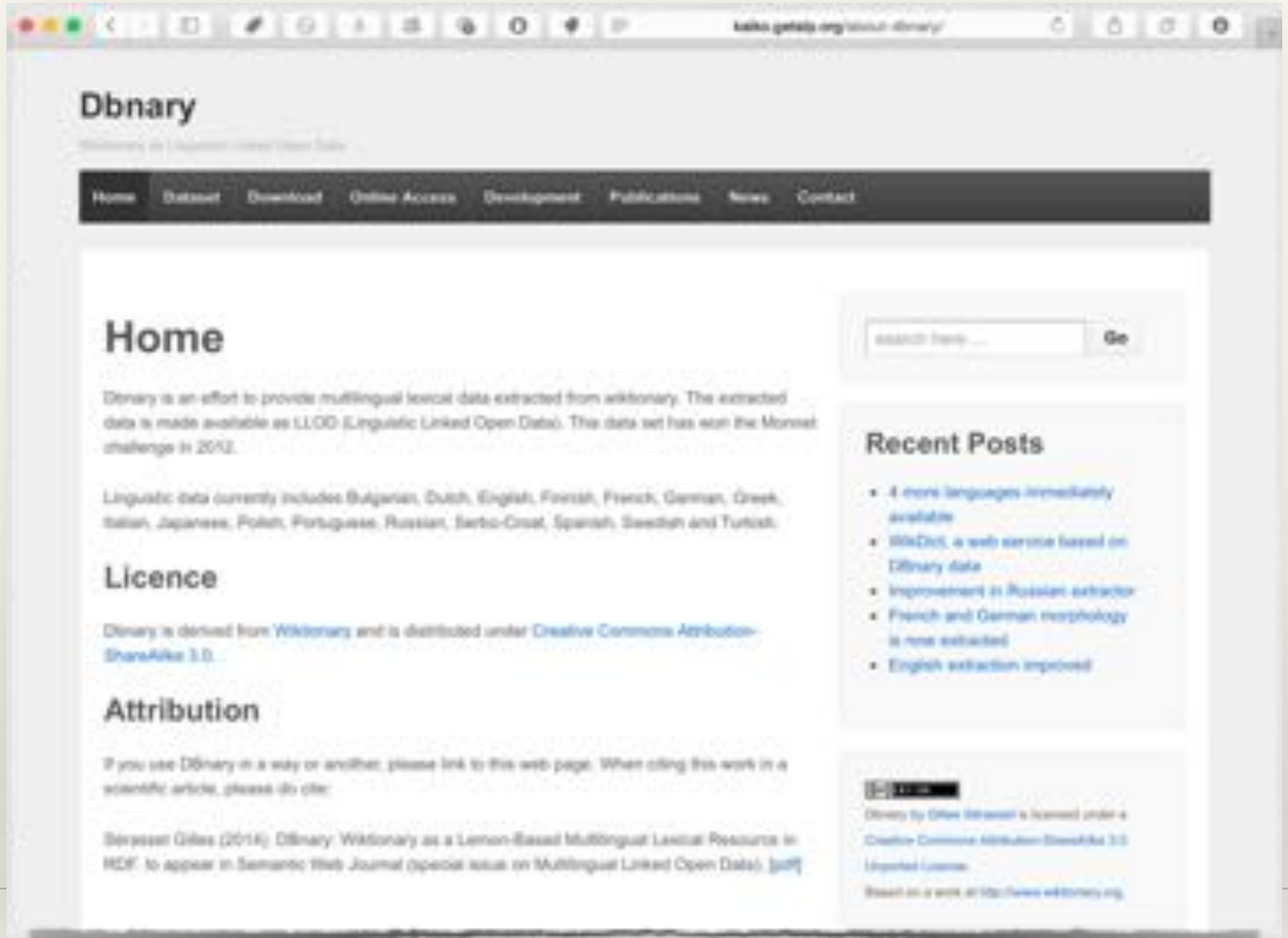


Size of the data



The Vauquois triangle





The DBnary website

Presentations, download, online access, tips and news

The DBnary website

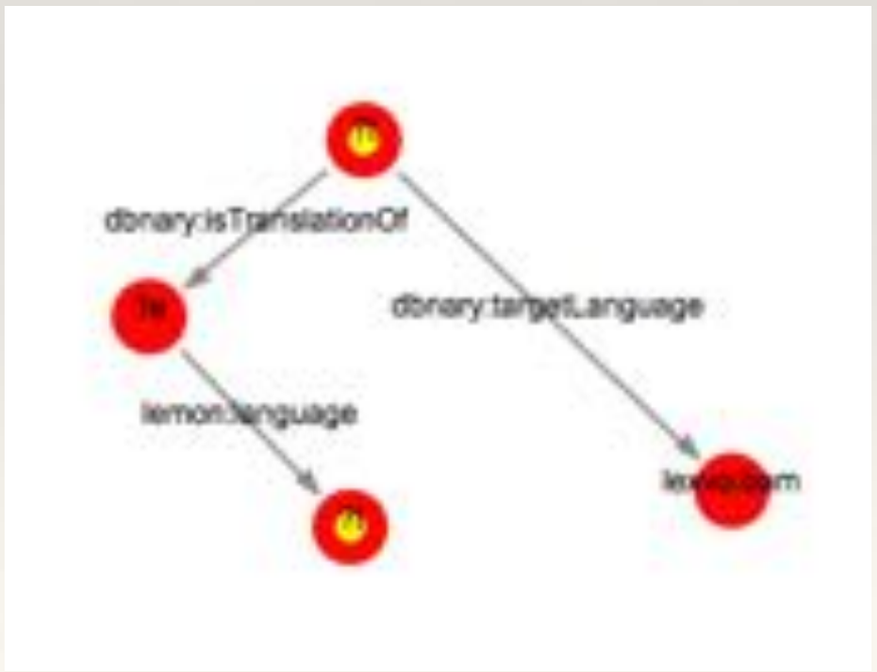
- http://kaiko.getalp.org/about-dbnary/
  - Dataset presentation, news, tips
- http://kaiko.getalp.org/fct
  - Facetted browser
- http://kaiko.getalp.org/sparql
  - Sparql endpoint
- http://kaiko.getalp.org/dbnary/fra/chat
  - Resolvable URIs for Linked Data (with content negociation)

The SPARQL endpoint

Example: counting translations to Bambara, grouped by source language

```
PREFIX lexvo: <http://lexvo.org/id/iso639-3/>
PREFIX dbnary: <http://kaiko.getalp.org/dbnary#>
PREFIX lemon: <http://www.lemon-model.net/lemon#>

SELECT count(?t) , ?l WHERE {
  ?t dbnary:targetLanguage lexvo:bam ;
  dbnary:isTranslationOf ?e .
  ?e lemon:language ?l
}
```



The SPARQL endpoint

- The extracted data is updated
  - AS SOON AS DUMPS ARE RELEASED
- Downloadable Turtle files are up to date
- The online data (fct, sparql) is usually out of sync with Turtle files





Lexsema a semantic swiss knife

Semantic similarities, unsupervised WSD with different similarity measures and meta-heuristics, supervised WSD,

Lexsema API

- A Java API for semantic similarity
  - A set of maven modules
    - Easily usable in your projects
    - still in alpha stage (quite reliable, but no documentation)
- developed mainly by Andon Tchechmedjiev

Lexsema API

org.getalp.lexsema-ontolex

1. Abstract API for the core model
2. SparQL query generation API
3. Generic core model querying (File, TDB, Remote endpoint)

org.getalp.lexsema-ontolex-dbnary

1. DBNary extended model
2. DBNary Querying

Lexsema API

org.getalp.lexsema-io

1. Load evaluation corpora (Semeval/Semcor, etc…)

2. Load lexical resource

3. Processing chain for raw text [DKPro/UIMA-fit]

4. Load gold standard data (WSD, CL-WSD)

5. Evaluate task results

6. Write task results

7. Preprocess and merge lexical resources (e.g. stemming definitions)

8. Generate dictionaries from resources

Lexsema API

org.getalp.lexsema-translation

1. Provide a unified API for MT services

2. Bing Translate Wrapper

3. Provide literal disambiguated target-word translations (dbnary cl-wsd)

org.getalp.lexsema-util

1. Caching API (REDIS, Memory)

2. DBNary Querying

Lexsema API

org.getalp.lexsema-similarity

1. Representation of abstract document structure (Text, Sentence, Word, Sense)

2. Generation of semantic signatures

3. Filtering of semantic signatures

4. Enrichment of semantic signatures (e.g. from Word2Vec model)

5. Semantic Similarity between senses in the same language

6. Semantic similarity between senses in different languages

Lexsema API

org.getalp.lexsema-ml

1. Function / Set Function API

2. Continuous optimization (e.g. Gradient Descent)

3. Score Matrix API

4. Generic Matrix Filtering

- FFT
- Value Normalization
- Dimensionality Reduction

5. Dimensionality Reduction

- Linear: PCA, SVD, FastICA, NeuralICA, NMF
- Non-linear: Tapkee Wrapper – LLE, KPCA, ISOMAP, MDE, etc

Lexsema API

org.getalp.lexsema-wsd

1. Word Sense Disambiguation API

2. Parameter Estimation

3. Sense raking regularization

4. Algorithms:

- Simplified Lesk
- Windowed Exhaustive Enumeration
- Simulated Annealing
- Genetic Algorithm
- Cuckoo Search
- Bat Search

Lexsema API

org.getalp.lexsema-wsd-supervised

1. Feature Extraction from Annotated corpora

2. Supervised Classification [Weka]

Example 1: Accessing DBnary

```
public static final String ONTOLOGY_PROPERTIES = "data" + File.separator + "ontology.properties”;
    public static void main(String[] args) throws […]{
        Store store = new JenaRemoteSPARQLStore("http://kaiko.getalp.org/sparql");
        StoreHandler.registerStoreInstance(store);
        OntologyModel model = new OWLTBoxModel(ONTOLOGY_PROPERTIES);
        DBNary dbnary = (DBNary) LexicalResourceFactory.getLexicalResource(DBNary.class, model, new Language[]
{Language.ENGLISH});
        Vocable v = null;
        try {
            v = dbnary.getVocable(args[0]);
            logger.info(v.toString());
            List<LexicalEntry> entries = dbnary.getLexicalEntries(v);
            for (LexicalEntry le : entries) {
                logger.info("\t" + le.toString());
                logger.info("\tRelated entities:");
                List<LexicalResourceEntity> related = dbnary.getRelatedEntities(le, DBNaryRelationType.synonym);
                for (LexicalResourceEntity lent : related) {
                    logger.info("\t\t" + lent.toString());
                }
                logger.info("\tSenses:");
                for(LexicalSense sense: dbnary.getLexicalSenses(le)){
                    logger.info("\t\t" + sense.toString());
                }
                logger.info("\tTranslations:");
                List<Translation> translations = dbnary.getTranslations(le);
                for (Translation translation : translations) {
                    logger.info("\t\t" + translation.toString());
                }
            }
        } catch (NoSuchVocableException e) {
            e.printStackTrace();
        }
    }
```

Example 1: Accessing DBnary

```
VocableImpl(vocable=cat, language=ENGLISH)
  ENGLISH LexicalEntry|cat#noun|
    Related entities:
      VocableImpl(vocable=guy, language=ENGLISH)
      VocableImpl(vocable=felid, language=ENGLISH)
       [...]
      VocableImpl(vocable=lion, language=ENGLISH)
    Senses:
      LexicalSense|__ws_1_cat__Noun__1|{'An animal of the family Felidae:'}
      LexicalSense|__ws_1a_cat__Noun__1|{'A domesticated subspecies (Felis silvestris
catus) of feline animal,

commonly kept as a house pet.'}
      LexicalSense|__ws_1b_cat__Noun__1|{'Any similar animal of the family Felidae, which
includes lions, tigers, bobcats, etc.'}
       [...]
      LexicalSense|__ws_9_cat__Noun__1|{'A vagina, a vulva; the female external
genitalia.'}
    Translations:
      TranslationImpl(gloss=domestic species@en, translationNumber=-1, writtenForm=gato,
language=PORTUGUESE)
      TranslationImpl(gloss=member of the family ''Felidae''@en, translationNumber=-1,
writtenForm=felino, language=PORTUGUESE)
      TranslationImpl(gloss=member of the subfamily Felinae@en, translationNumber=-1,
writtenForm=Kleinkatze, language=GERMAN)
      TranslationImpl(gloss=member of the family ''Felidae''@en, translationNumber=-1,
writtenForm=felina, language=CATALAN)
language=FINNISH)
```

Example 2: Text similarity

```
public final class TextSimilarity {
    private static Logger logger = LoggerFactory.getLogger(TextSimilarity.class);

    public static void main(String[] args) {

        if (args.length <2) {
            usage();
        }
        StringSemanticSignature signature1 = new StringSemanticSignatureImpl(args[0]);
        StringSemanticSignature signature2 = new StringSemanticSignatureImpl(args[1]);

        SimilarityMeasure similarityMeasure = new TverskiIndexSimilarityMeasureBuilder()
                .alpha(1d).beta(0).gamma(0).computeRatio(false).fuzzyMatching(false).normalize(true).r
egularizeOverlapInput(true).build();
        double sim = similarityMeasure.compute(signature1, signature2);
        String output = String.format("The similarity between \"%s\" and \"%s\" is %s",
signature1.toString(), signature2.toString(), sim);
        logger.info(output);
    }

    private static void usage() {
        logger.error("Usage -- TextSimilarity \"String1\" \"String2\"");
        System.exit(1);
    }
}
```

Example 2: Text similarity

```
java –jar org.getalp.lexsema-examples-allinone.jar \
                org.getalp.lexsema.examples.TextSimilarity "The king is dead" "Long live the king”
```

```
 INFO [main] (TextSimilarity.java:25) - The similarity between " The king is dead" and " Long
live the king" is 0.25
```

Example 3: Crosslingual text similarity

```
public class CrossLingualTextSimilarity {
    private static Logger logger = LoggerFactory.getLogger(CrossLingualTextSimilarity.class);

    public static void main(String[] args) throws IOException, InvocationTargetException,
NoSuchMethodException, ClassNotFoundException, InstantiationException, IllegalAccessException {
        if(args.length<4){
            usage();
        }
        Language source = Language.fromCode(args[0]);
        Language target = Language.fromCode(args[1]);
        StringSemanticSignature signature1 = new StringSemanticSignatureImpl(args[2]);
        signature1.setLanguage(source);
        StringSemanticSignature signature2 = new StringSemanticSignatureImpl(args[3]);
        signature2.setLanguage(target);
        Translator translator = new GoogleWebTranslator();
        SimilarityMeasure similarityMeasure = new TverskiIndexSimilarityMeasureBuilder()
                .fuzzyMatching(true).alpha(1d).beta(0d).gamma(0d).normalize(true).regularizeOverlapIn
put(true).build();
        SimilarityMeasure crossLingualMeasure = new
TranslatorCrossLingualSimilarity(similarityMeasure, translator);
        double sim = crossLingualMeasure.compute(signature1, signature2);
        String output = String.format("The similarity between \"%s\" and \"%s\" is %s",
signature1.toString(), signature2.toString(), sim);
        logger.info(output);
    }
 }
```

Example 3: Translingual text similarity

```
java –jar org.getalp.lexsema-examples-allinone.jar \
                        org.getalp.lexsema.examples.CrossLingualTextSimilarity en fr \
                        "Long live the king" "Longue vie au roi"
```

```
 INFO [main] (CrossLingualTextSimilarity.java:40) - The similarity between "
Long live the king" and " Longue vie au roi" is 1.0
```

Example 4: WSD

```
// Load and segment the text
TextLoader textLoader = new RawTextLoader(new StringReader(args[0]), new EnglishDKPTextProcessor());

// Connect to DBnary
Store store = new JenaRemoteSPARQLStore("http://kaiko.getalp.org/sparql");
StoreHandler.registerStoreInstance(store);
OntologyModel model = new OWLTBoxModel(ONTOLOGY_PROPERTIES);
DBNary dbnary = (DBNary) LexicalResourceFactory.getLexicalResource(DBNary.class, model, new
Language[] {Language.ENGLISH});

        LRLoader lrloader = new DBNaryLoaderImpl(dbnary, Language.ENGLISH).loadDefinitions(true);

        // Define similarity measure
        SimilarityMeasure similarityMeasure =
                new TverskiIndexSimilarityMeasureBuilder()
                        .distance(new ScaledLevenstein()).alpha(1d).beta(0.0d).gamma(0.0d)
                        .fuzzyMatching(true)
                        .build();
        ConfigurationScorer configurationScorer =
                new ConfigurationScorerWithCache(similarityMeasure);

        // Choose a meta heuristic for WSD
        Disambiguator disambiguator = new CuckooSearchDisambiguator(1000,.5,1,0,configurationScorer,true);
```

Example 4: WSD

```
System.err.println("Loading texts");
textLoader.load();
for (Document document : textLoader) {
    System.err.println("\tLoading senses...");
    lrloader.loadSenses(document);
    System.err.println("\tDisambiguating... ");
    Configuration result = disambiguator.disambiguate(document);
    for(int i=0;i<result.size();i++){
        Word word = document.getWord(i);
        if(!document.getSenses(i).isEmpty()) {
            String sense = document.getSenses(i).get(result.getAssignment(i)).getId();
            logger.info("Sense " + sense + " assigned to " + word);
        }
    }

    System.err.println("done!");
}
disambiguator.release();
```

Example 4: WSD

```
java -jar org.getalp.lexsema-examples-allinone.jar \
     org.getalp.lexsema.examples.TextDisambiguation \
    "I would like to take some time to congratulate you for your victory"
```

```
INFO [main] (TextDisambiguation.java:68) - Sense LexicalSense|__ws_9g_take__Verb__1|{'To accept,
as something offered; to receive; not to refuse or reject; to admit.'} assigned to ENGLISH
LexicalEntry|take#verb|

INFO [main] (TextDisambiguation.java:68) - Sense LexicalSense|__ws_2f_time__Noun__1|{'A person's
youth or young adulthood, as opposed to the present day.'} assigned to ENGLISH LexicalEntry|
time#noun|

INFO [main] (TextDisambiguation.java:68) - Sense LexicalSense|__ws_1_congratulate__Verb__1|{'to
express one’s sympathetic pleasure or joy to the person(s) it is felt for'} assigned to ENGLISH
LexicalEntry|congratulate#verb|

INFO [main] (TextDisambiguation.java:68) - Sense LexicalSense|__ws_1_victory__Noun__1|{'An
instance of having won a competition or battle.'} assigned to ENGLISH LexicalEntry|victory#noun|
```



To infinity and beyond!

Axies

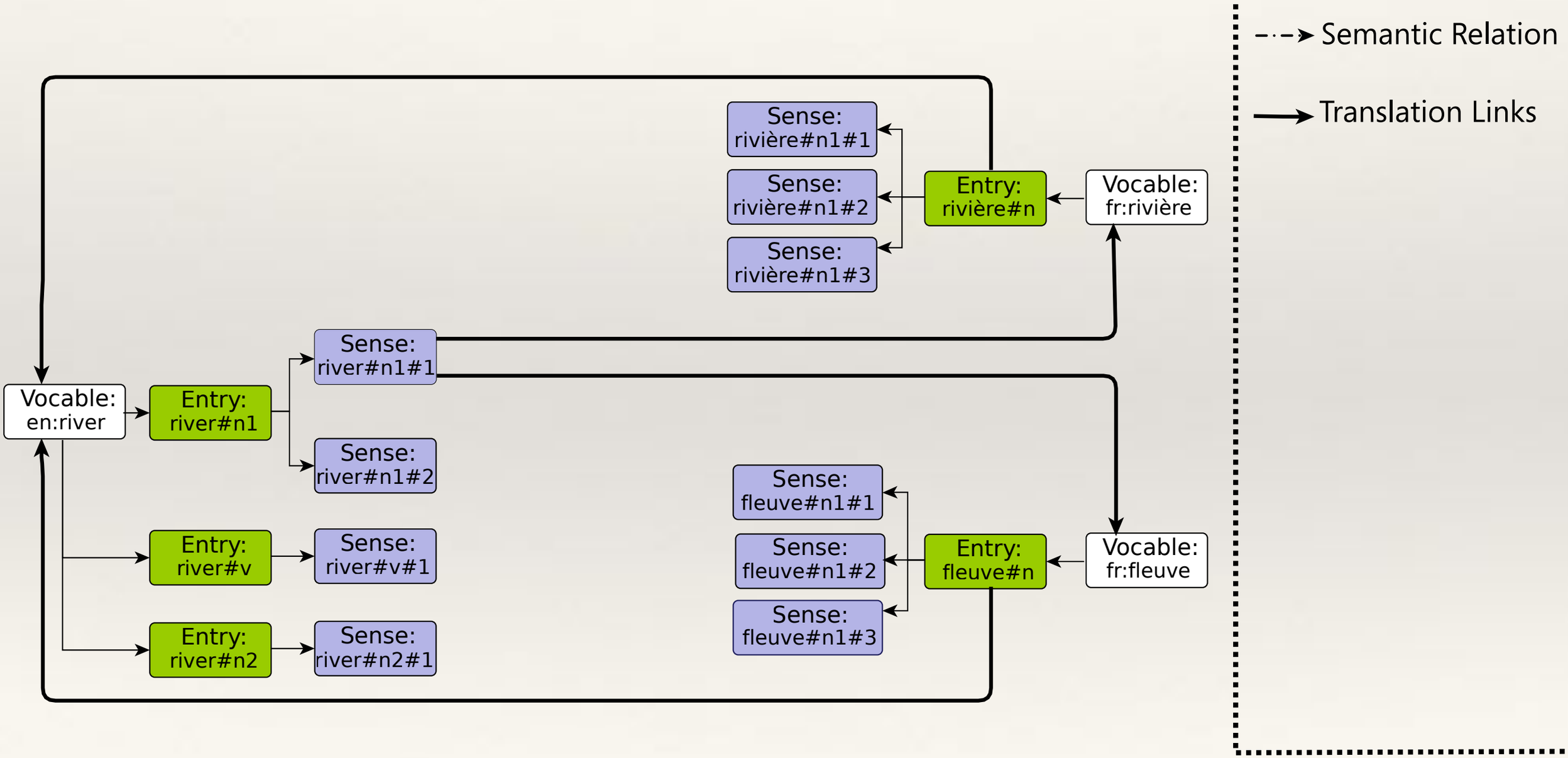
The corner stone towards linguistic felicity in pivot based multilingual lexical resources

Axies = Interlingual acceptions

- A lexical architecture defined in 1994
  - G. Sérasset. Interlingual Lexical Organisation for Multilingual Lexical Databases in NADIA. In M. Nagao, editor, COLING-94, volume 1, pages 278–282, August 1994.
- “Interlingual acceptions” —> Axies
  - A simpler name for the same concept in the “Papillon” project.

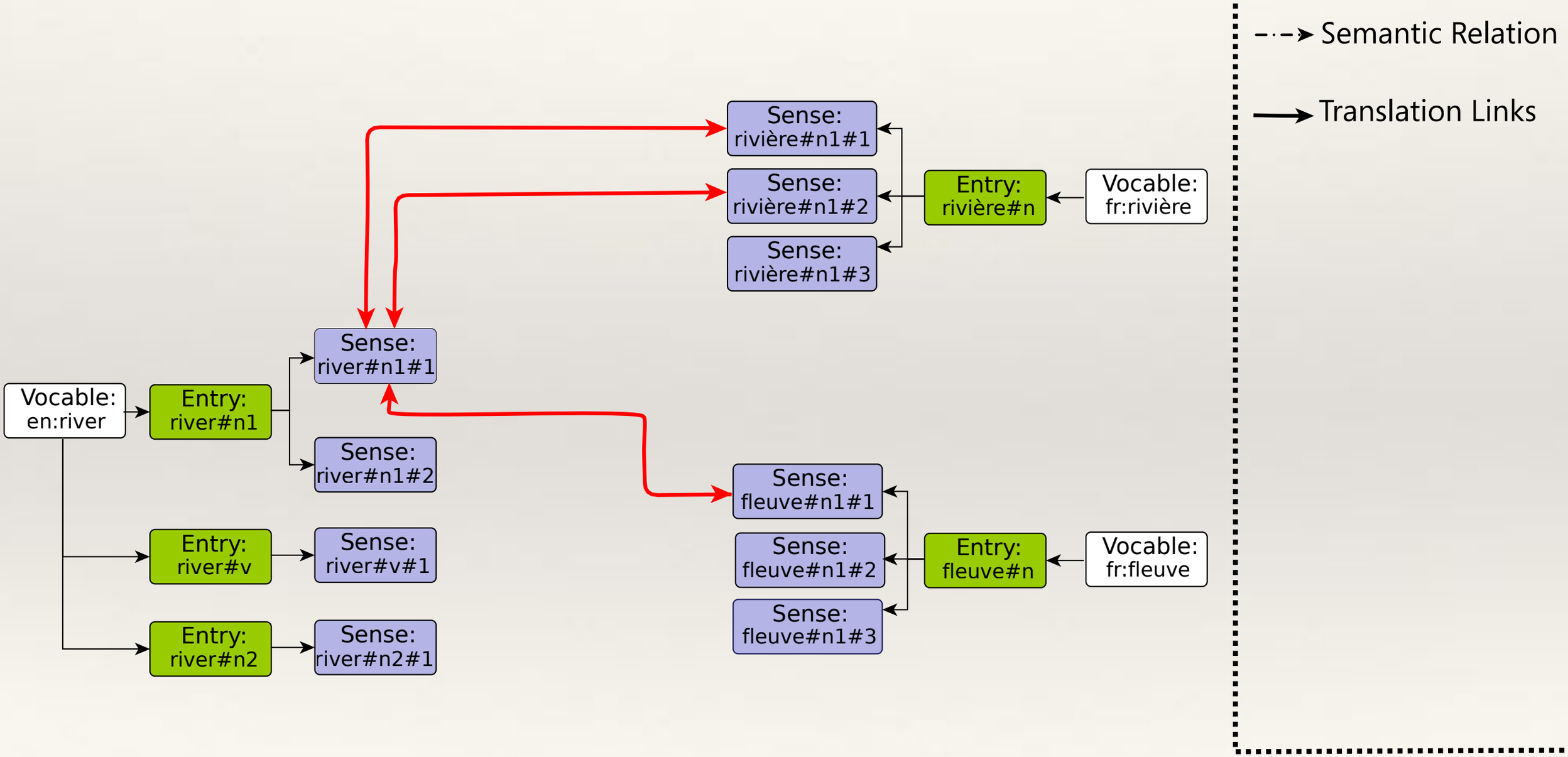
Axies: illustration

Existing multilingual data (multi-bilingual dictionaries)



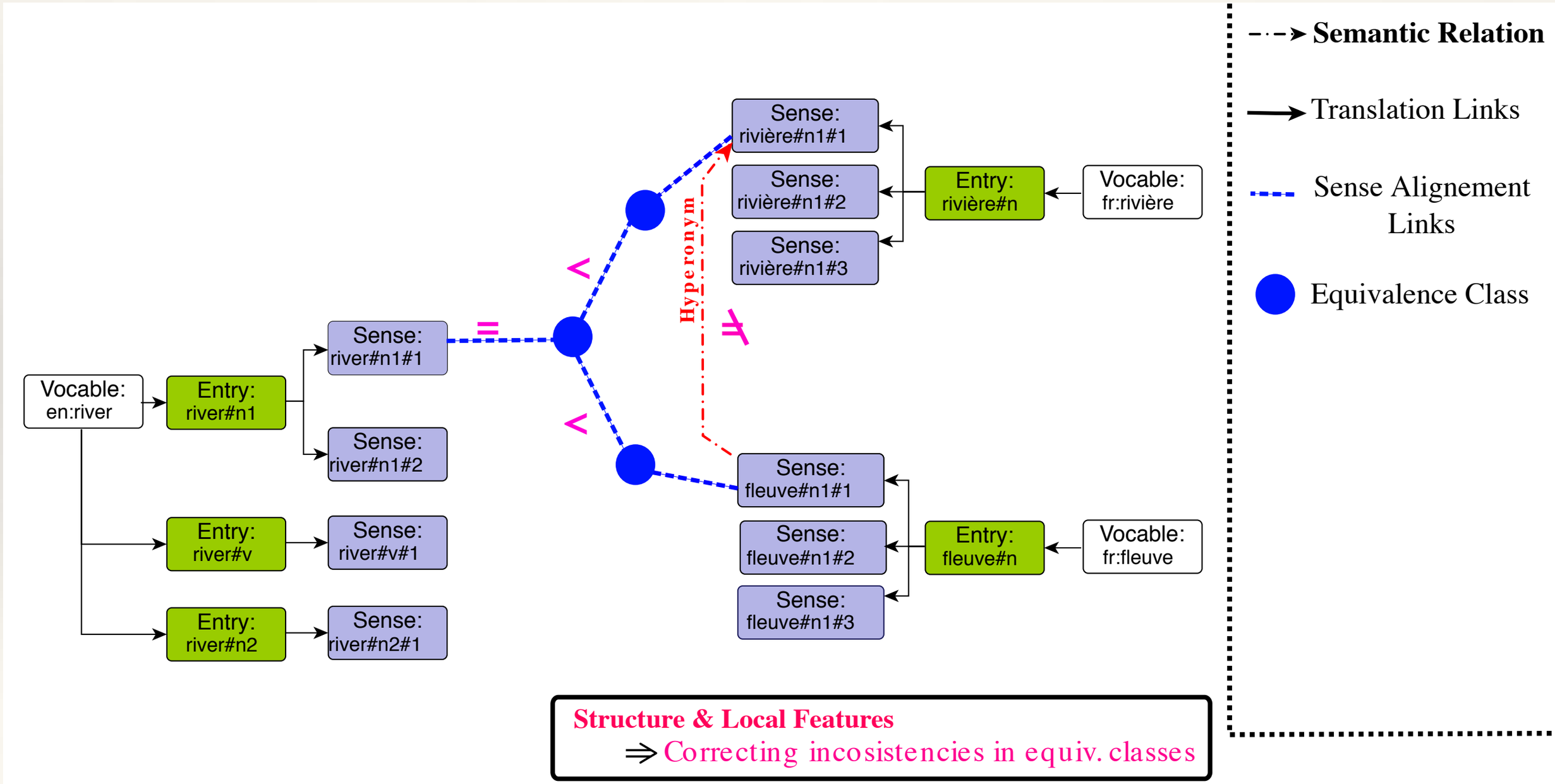
Axies: illustration

Better non existing multilingual data (multi-bilingual bidirectional dictionaries)



Axies: illustration

Even better multilingual data (multilingual dictionaries)



Axies based lexical resources ?

- Papillon project
  - tentative building of an axis based LR
  - —> hard to do when many different languages
    - the data itself is beyond each expert knowledge
- Andon's thesis
  - Create this resource semi automatically from bilingual data
  - —> An axie is the equivalence class of an equivalence relation that is not known but that may be partially observed in existing bilingual LRs.

Lexsema API

org.getalp.lexsema-acceptali

1. Sense Closure Computation

2. Aligning Senses Across Languages

3. Clustering Word Senses

