You shall find the target via its companion words…
In search for a model for interconnecting lexicographic resources

Michael Zock[1] and Dan Cristea[2,3]

[1] Aix-Marseille Université, CNRS

[2] "Alexandru Ioan Cuza" University of Iași

[3] Institute of Computer Science, Romanian Academy

michael.zock@lif.univ-mrs.fr, dcristea@info.uaic.ro

What features are important in the TOT search?

- There should be a word (or more) to start the search from
- Michael's *expansion-clustering* model is followed recursively
- The *expansion* step occurs if a word triggers more words => the resource should support the expansion
- *Clustering* could be done if features exist
- Search should use a resource or a combination of resources

The first thought: standardisation

- Lexical Markup Framework (LMF)
  - What is it?
    - a common model for creation and use of lexical resources
  - With what goal?
    - to manage the exchange of data between and among these resources
    - to enable the merging of a large number of individual electronic resources to form extensive global electronic resources

Near-standard

- Text Encoding Initiative (TEI)
  - What is it?
    - an inventory of the features most often deployed for computer-based text processing
    - recommendations about suitable ways of representing these features
  - With what goal?
    - to facilitate processing by computer programs
    - to facilitate the loss-free interchange of data amongst individuals and research groups using different programs, computer systems, or application software

Standardisation

- Text Encoding Initiative (TEI)
  - Example of a dictionary entry serialisation

    (from TEI Guidelines)

disproof (dIs"pru:f) n. 1. facts that disprove something. 2. the act of disproving. CED

```
<entry>
  <form>
    <orth>disproof</orth>
    <pron>dIs"pru:f</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <sense n="1">
    <def>facts that disprove something.</def>
  </sense>
  <sense n="2">
    <def>the act of disproving.</def>
  </sense>
</entry>
```

Needs

- If I want to connect two resources, simply merge the contents
- Then query the merged resource by taking advantage of peculiarities in each resource
- For querying use classical database or semantic tools (as relational operators or RDF inference)

Needs

- *Expansion* by repeatedly using the same resource
  - Give me the `definition` neighbouring sphere of depth 2 of the word *captain* (take all senses of the entry *captain* and form the list of words in the corresponding definitions, then for each of them take all their senses and collect again words in their definitions).

Needs

- *Expansion* by combining more resources
  - I want all lemmas appearing in contexts of words belonging to citations of the entry *symphony*.

corpus

dictionary

Parameterising the needs

1. **Representation**: directed and connected graph
   – nodes: feature structures (complex data) or values (words, definitions, etc.)
   – edges: named relations (e.g. lemma, morphological data, word senses and citations for **dictionary entries**, sentence id and contextual POS for words in **corpora**, etc.)

Parameterising the needs

2. **Completeness**: gives an estimation of the size of the resource

- aim: retrieve any word of a language => resource should include as many of its words
- property evaluated in fuzzy terms, because no resource is complete (for instance, no proper nouns, newly coined terms, obsolete words, etc.)

Parameterising the needs

3. **Features**: resources should be characterised by a rich collection of features
   - to be used both in *expansion* and *clustering*
   - features will help to spread activation from the spotted word in the *expansion* step and as criteria for *clustering*

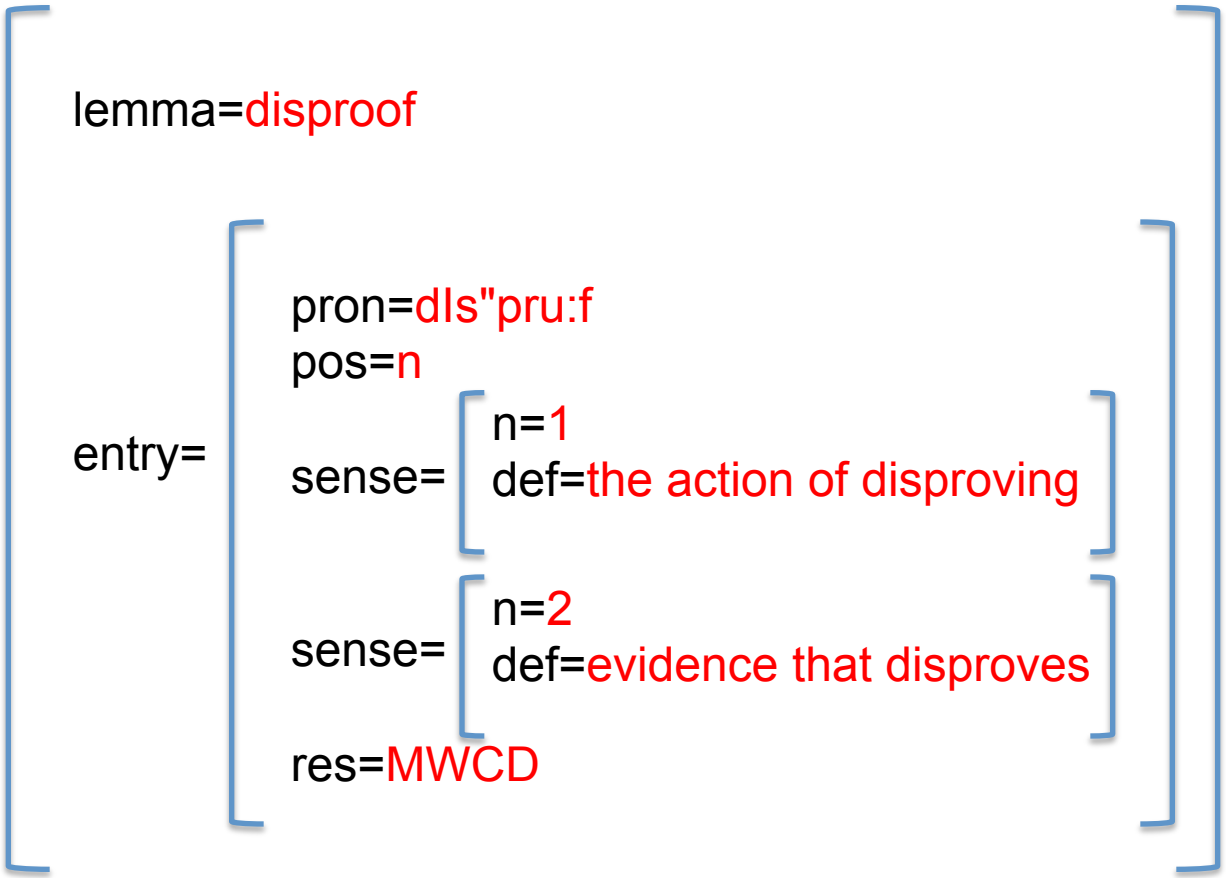
A bunch of notorious resources: an explanatory dictionary

1. *Representation*: directed graph
   – a collection of entries

2. *Completeness*: close to 100%
   - but it has no proper nouns

3. *Features*: POS, LEMMA, SENSE, CITATION, etc.

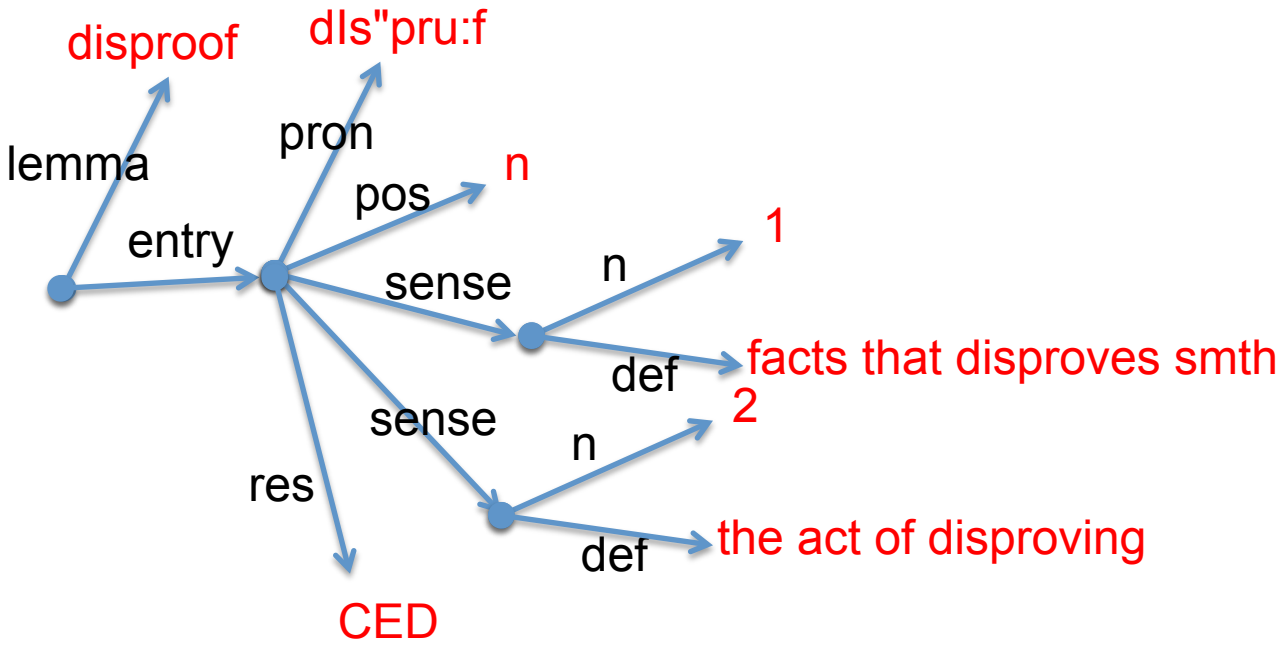
Representing lexical entries as feature structures



Merriam-Webster's Collegiate Dictionary

Representing lexical entries as directed graphs

- Graph representation



A bunch of notorious resources: WordNet

1. *Representation*: directed graph
   - resource split in 4: nouns, verbs, adjectives and adverbs, but connectivity assured by direct access

2. *Completeness*: perhaps enough for rich WNs
   - but usually very few proper nouns

3. *Features*: POS, LEMMA, SENSE
   - but also semantic relations: HYPERNYMY, HYPONYMY, ANTONYMY, etc.

The WordNet search for *discount*

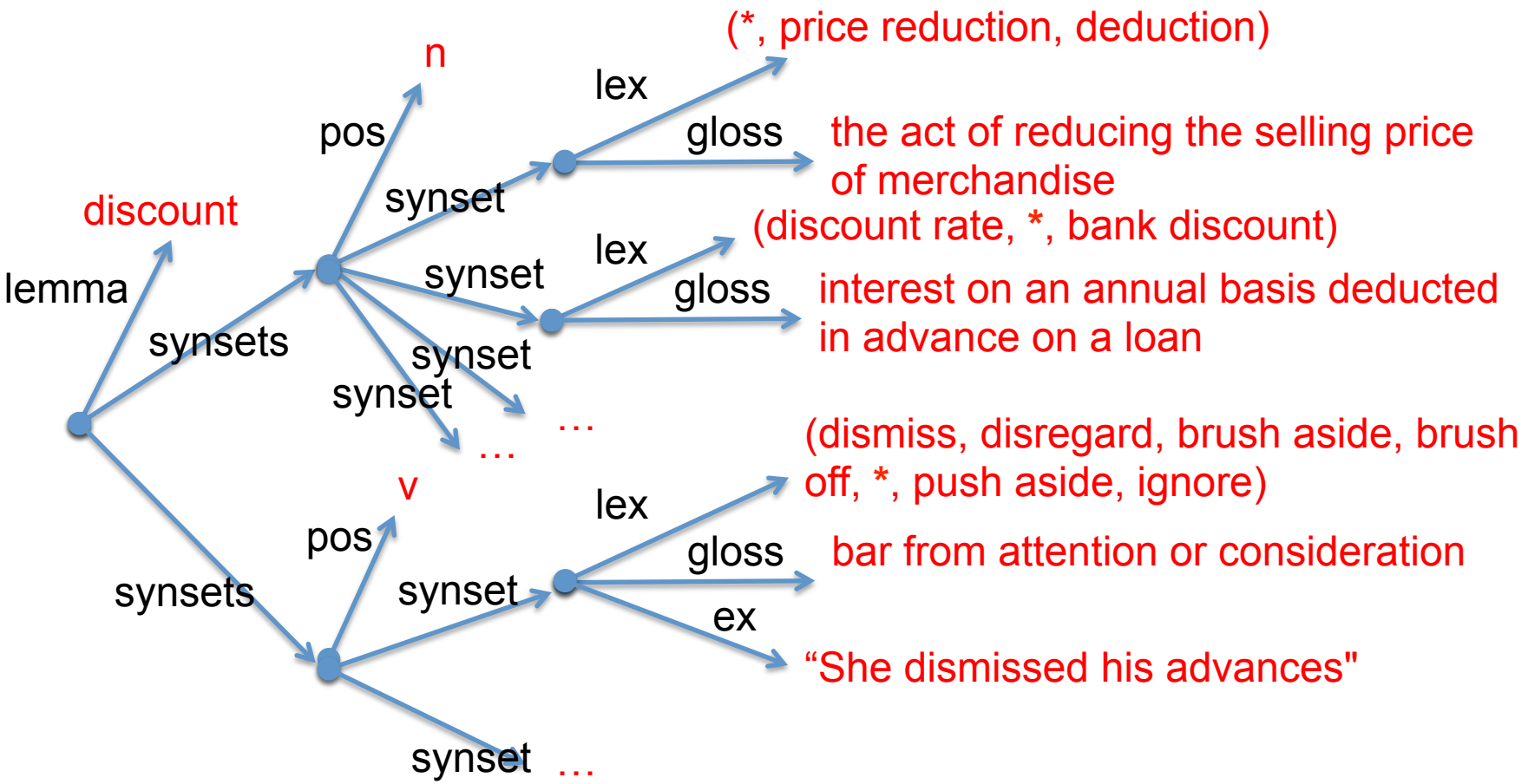
Noun

- S: (n) **discount**, price reduction, deduction (the act of reducing the selling price of merchandise)
- S: (n) discount rate, **discount**, bank discount (interest on an annual basis deducted in advance on a loan)
- S: (n) rebate, **discount** (a refund of some fraction of the amount paid)
- S: (n) deduction, **discount** (an amount or percentage deducted)

Verb

- S: (v) dismiss, disregard, brush aside, brush off, **discount**, push aside, ignore (bar from attention or consideration) *"She dismissed his advances"*
- S: (v) **discount** (give a reduction in price on) *"I never discount these books–they sell like hot cakes"*

Representing WordNet synsets



Notorious resources:
a corpus

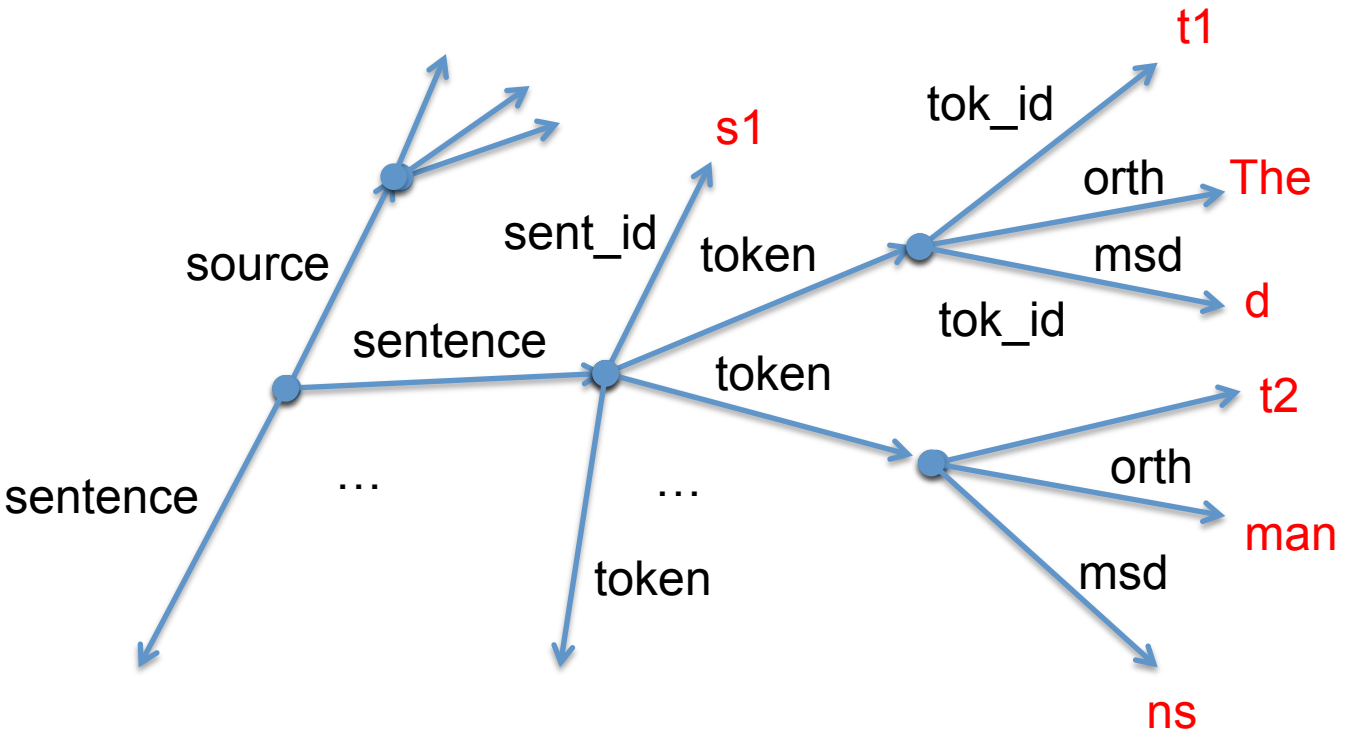
1. *Representation*: a collection of sentences represented as a graph

2. *Completeness*: usually extremely large
   - includes also proper nouns

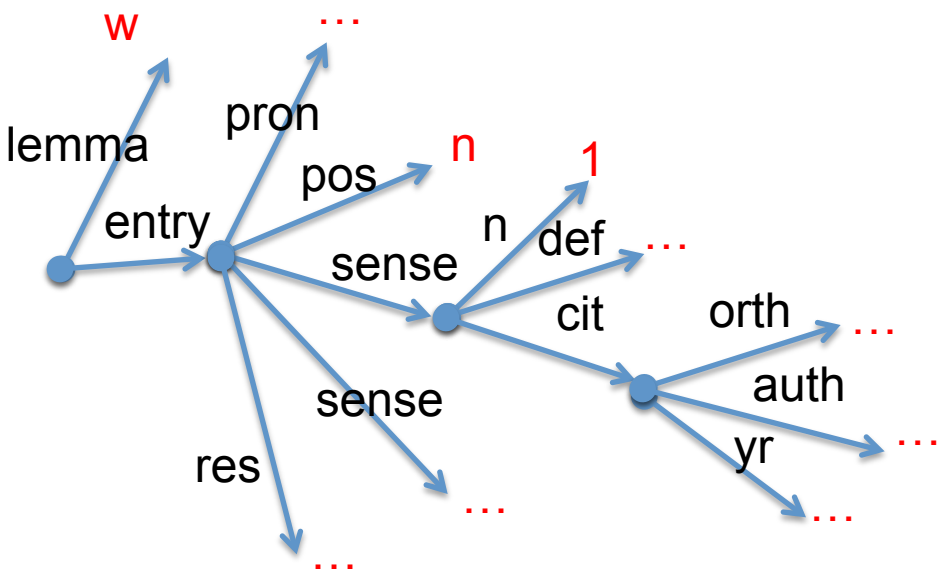
3. *Features*: SENTENCE, TOKENS, POS, even SENSE or SYNTACTICAL structure, etc.

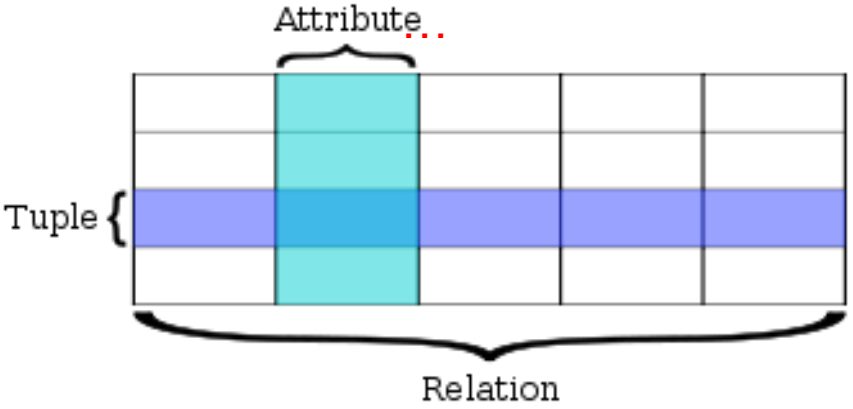
Representing a corpus as a directed graphs

- Graph representation



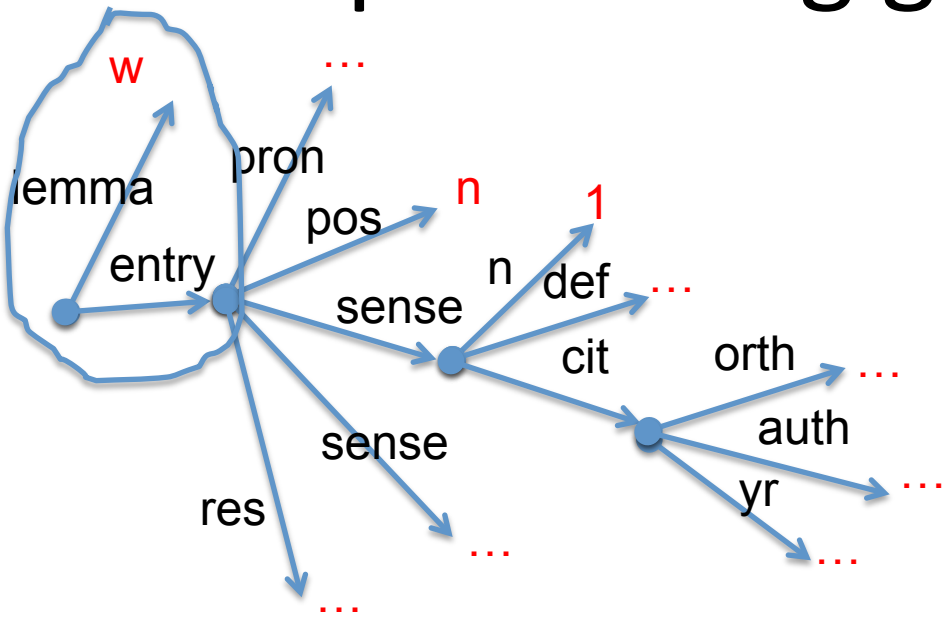
One step further: representing graphs as tables



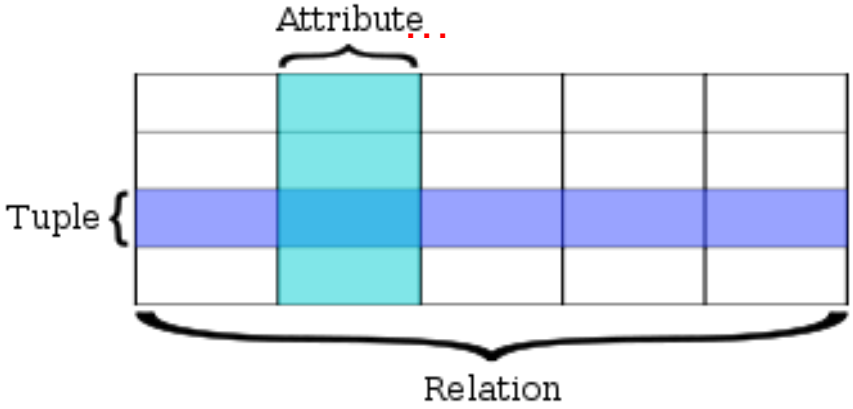


from http://en.wikipedia.org/wiki/Relational_database

One step further: representing graphs as tables

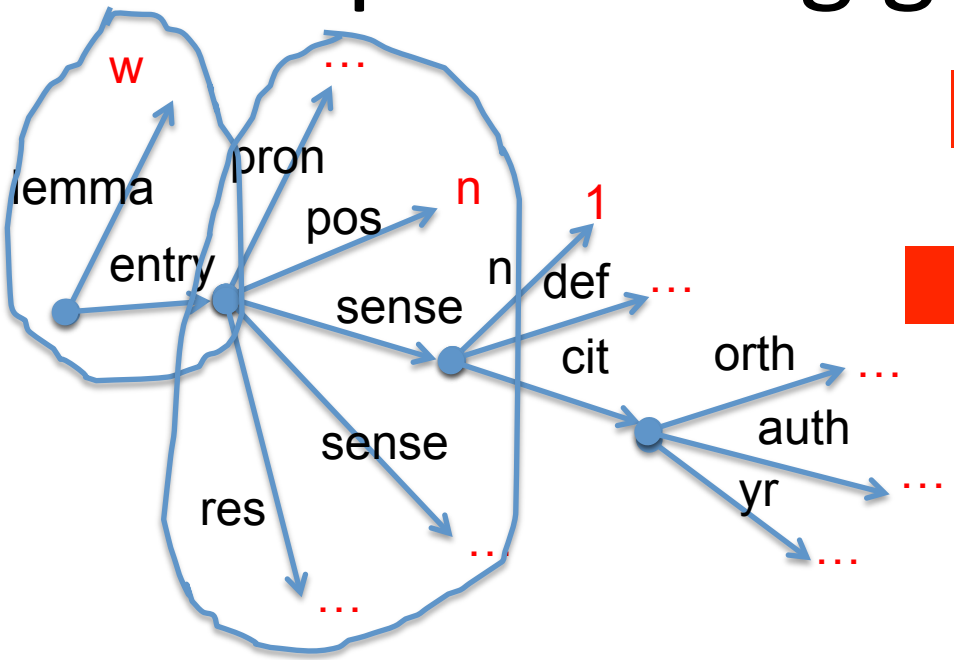


WORD	id	lemma	entry



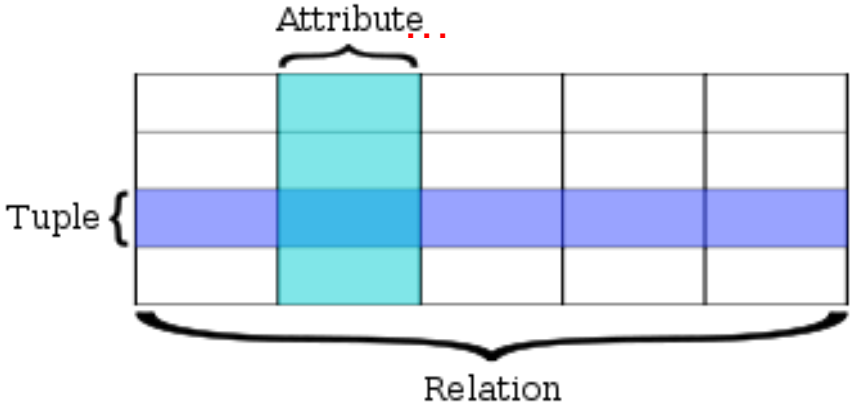
from http://en.wikipedia.org/wiki/Relational_database

One step further: representing graphs as tables



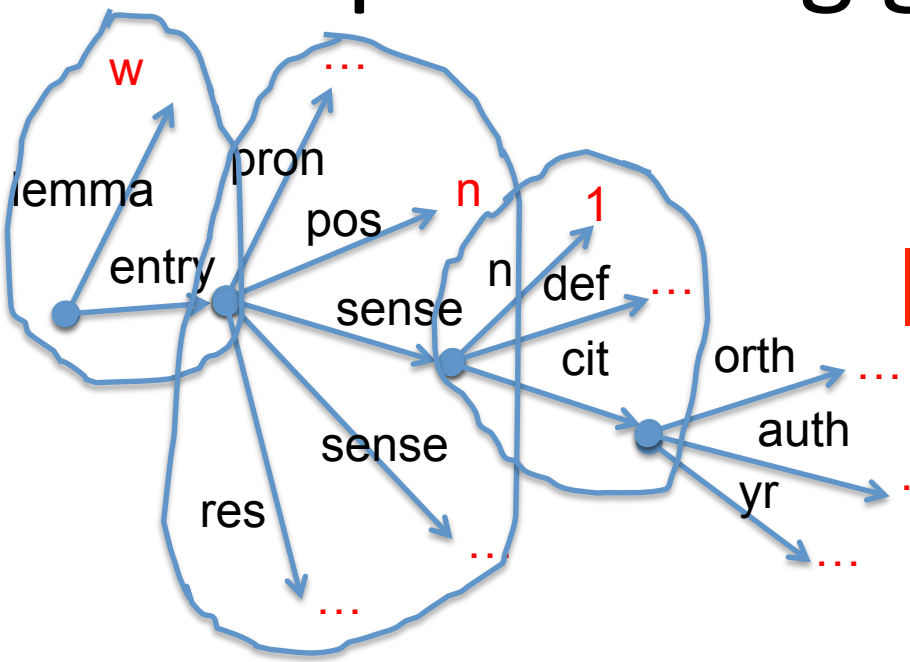
WORD	id	lemma	entry

ENTRY	entry	pron	pos	sense	res

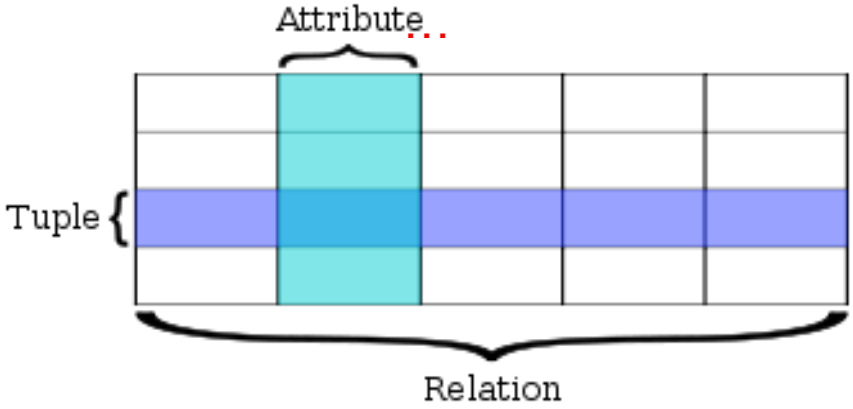


from http://en.wikipedia.org/wiki/Relational_database

One step further: representing graphs as tables

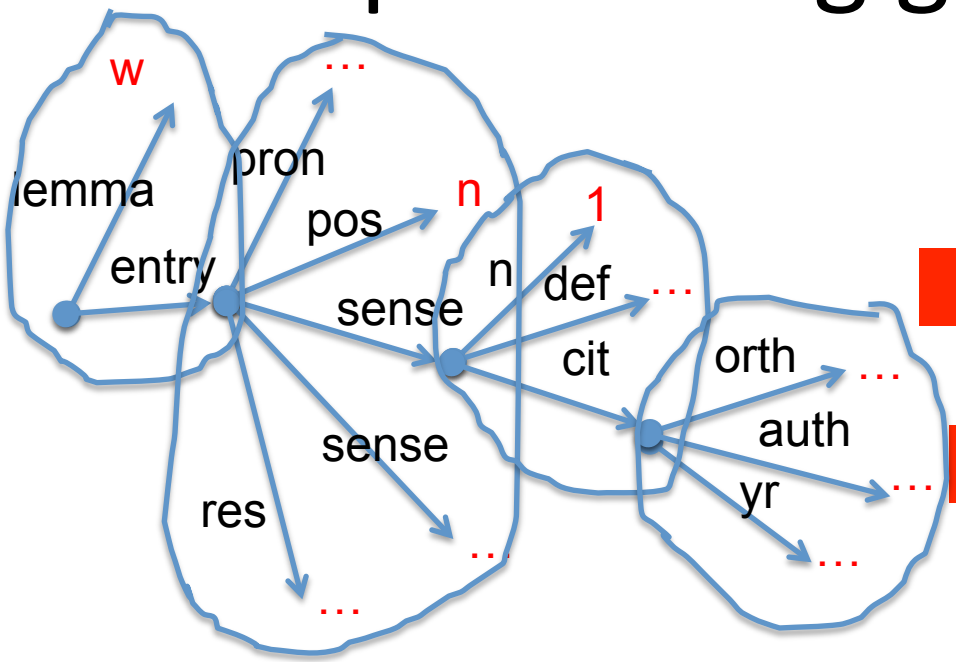


WORD	id	lemma	entry
ENTRY	entry	pron	pos	sense	res
SENSE	sense	n	def	cit



from http://en.wikipedia.org/wiki/Relational_database

One step further: representing graphs as tables

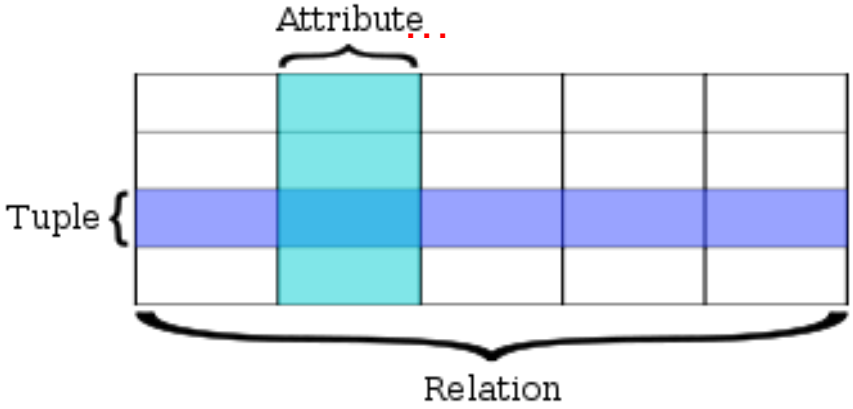


WORD	id	lemma	entry

ENTRY	entry	pron	pos	sense	res

SENSE	sense	n	def	cit

CIT	cit	orth	auth	yr



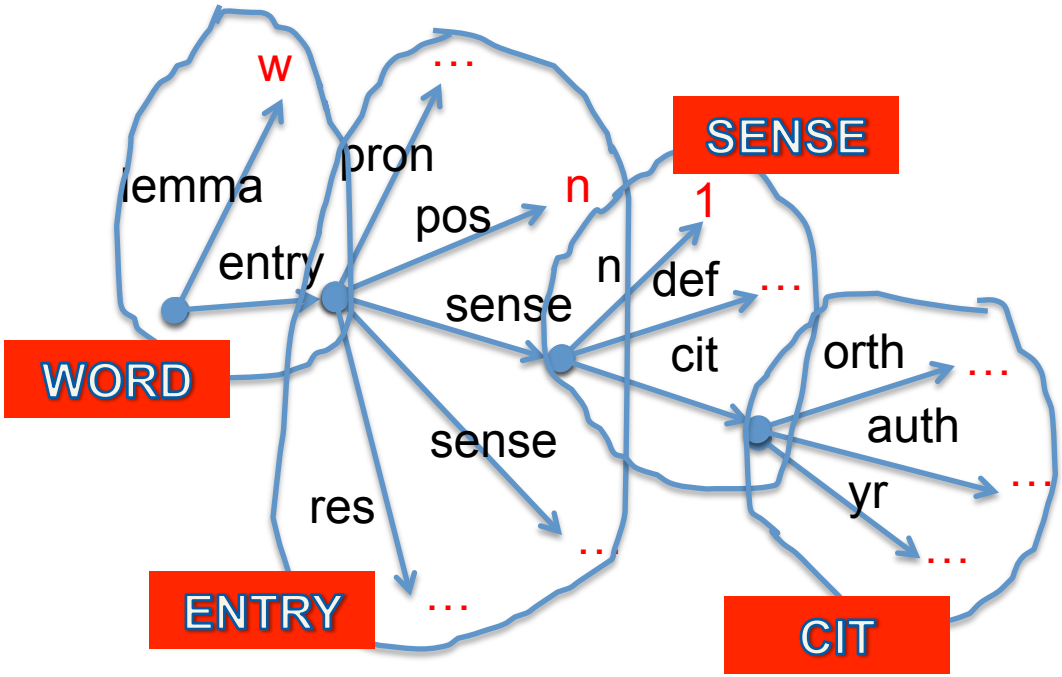
from http://en.wikipedia.org/wiki/Relational_database

Relational operators

- **Projection**: $\pi_{a1,\ldots an}(R)$ => a relation containing only values of attributes a1,… an from the relation R
- **Selection**: $\sigma_{\phi}(R)$, with φ is logical condition => only tuples verifying the condition φ are retained from the relation (or the set) R
- **Join**: R✜S => the set of all attributes in R and S that are equal on their common attributes

An *expansion* sphere of depth 1 in a dictionary

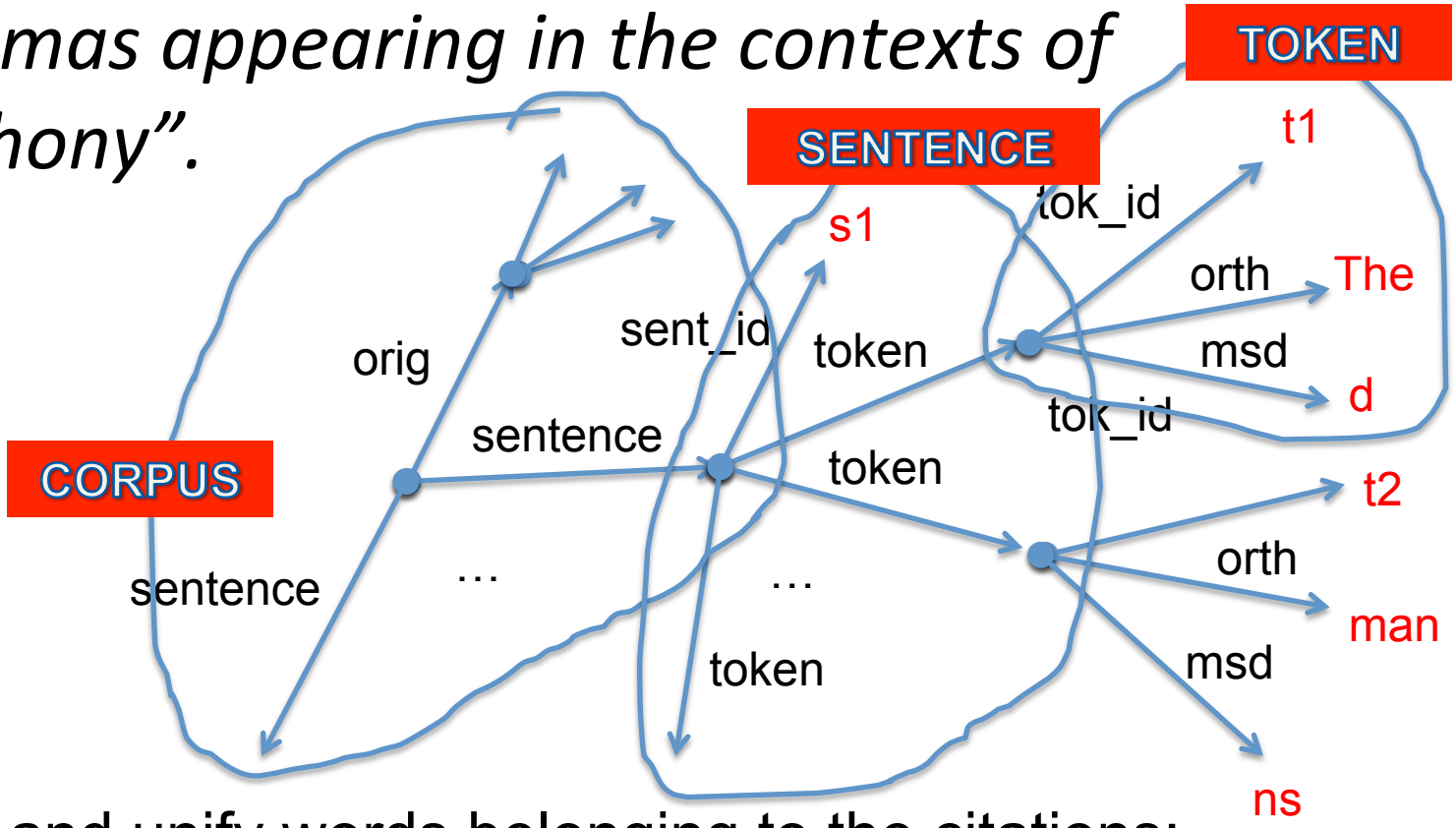
- *All lemmas in the citations of the entry "symphony".*



Lemmatise and unify words belonging to the citations:
$U(lem(\pi_{orth}(\sigma_{lemma=\text{“symphony”}}(WORD \clubsuit ENTRY \clubsuit SENSE \clubsuit CIT))))$

An *expansion* sphere of depth 1 in a corpus

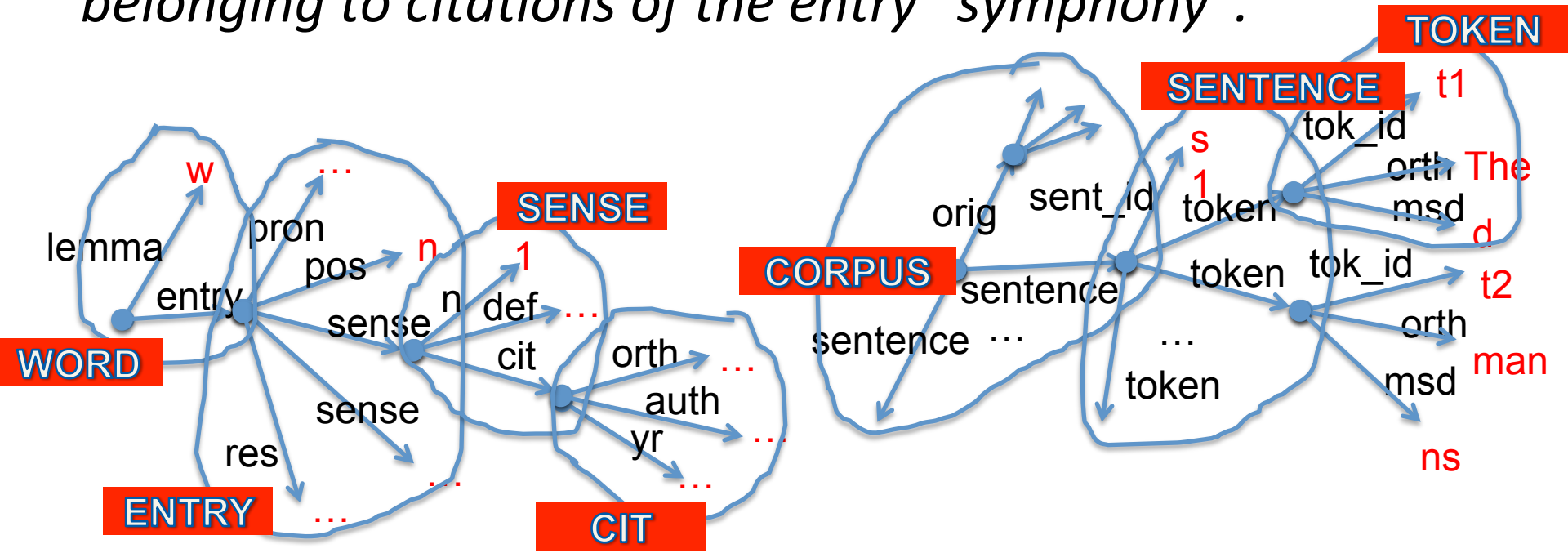
- *All lemmas appearing in the contexts of "symphony".*



Lemmatise and unify words belonging to the citations:
$U(lem(\pi_{orth}(\sigma_{orth=\text{“symphony”}}(SENTENCE \text{✣} TOKEN))))$

An *expansion* sphere of depth 2 by connecting a dictionary and a corpus

- *All lemmas appearing in contexts of words belonging to citations of the entry "symphony".*



Lemmatise and unify words belonging to the citations:

$U(lem(\pi_{orth}(\sigma_{orth \in U(lem(\pi orth(\sigma lemma=\text{"symphony"}\ (WORD \clubsuit ENTRY \clubsuit SENSE \clubsuit CIT))))}\ (SENTENCE \clubsuit TOKEN)))$

Conclusions

- We discussed here ideas to implement the *expansion* step of Michael's 2 steps TOT model
- Central: standardize and link lexicographic resources of different types
  - resource => TEI representation => as feature structures => relational tables (or RDF tuples)
  - query by using relational operators (or RDF inference)

Discussion

- Only a sketch
  - a lot of details should still be filled in: *clustering*
- The good news:
  - XML structures (the native language of TEI) accept direct representations as database records: XSLT => opening direct access to a complex querying language: XQuery => RDF reasoning
- A handy tool:
  - interrogations can be pre-formulated

Acknowledgements

- Work partially supported by the project *The Computational Representative Corpus of Contemporary Romanian Language*, a project of the Romanian Academy and partially by the COST-ENeL project

Thank you!