How to Database Meaning: Building a Typological Database of Temperature Terms

by

Susanne Vejdemo

Thesis

Submitted to the Department of English Language and Literature

Eastern Michigan University

In partial fulfillment of the requirements

For the degree of

MASTER OF ARTS

in

English Linguistics

Thesis Committee:

Anthony Aristar

Helen Aristar-Dry

November 22, 2013

Ypsilanti, Michigan

Acknowledgments

I will always be grateful to Professors Anthony Aristar and Helen Aristar-Dry at the Institute for Language Information and Technology, EMU, for their advice and teaching – it has been a privilege to have the chance to learn from them.

This thesis would not have been possible without the continuous support and guidance from my mentor, Professor Maria Koptjevskaja Tamm, Stockholm University.

Abstract

The purpose of this master's thesis is to discuss the design and use of databases and database interfaces for lexical semantic research in general, and the design of the Typological Database of Temperature Terms (TDTT) in particular. It chronicles the design process of the TDTT, from use cases to conceptual and logical design, and finally to physical implementation. The database design will be contrasted to other database resources in lexical semantics research, such as Wordnet and Framenet, and two different approaches to the physical database model will be analyzed: RDF and SQL. The thesis concludes with a discussion of some of the conclusions that can be drawn about database design for small scale lexical semantic databases.

Table of Contents

List of Figures

Figure Page

List of Tables

Table Page

Chapter 1: Introduction

The purpose of this master's thesis is to discuss the design and use of databases and database interfaces for lexical semantic research in general and the design of the Typological Database of Temperature Terms (TDTT) in particular.

I intend to describe the design process of the TDTT, from use cases to conceptual and logical design and finally to physical implementation. I will discuss in depth the motivation behind the different design choices.

The thesis will begin with a short introduction to lexical typology and temperature term research, followed by a discussion of the general requirements and use cases that any research database focusing on lexical semantics should address. I will then discuss several existing databases that are frequently used in lexical semantic research (including variations of Wordnet and Framenet), and how well they handle these requirements. This is followed by a section on the relative merits of a physical database model using SQL, compared to RDF, in light of the particular needs of the research field.

I then turn to the design of the TDTT and the specific use cases that typological temperature term research entails. Once these are presented, I proceed to sketch out both an RDF and an SQL version of the database and discuss their respective challenges and virtues. I explain why the SQL version is the most promising for this particular research project and proceed to describe its physical implementation. Since a partial goal of this thesis is to focus on the design process, not just the ultimate design, of lexical semantic databases, I will then discuss the problems encountered with the first SQL implementations, once data sets from 33 languages were added to the database. These problems led to substantial redesigns of parts of the database.

The last section discusses some of the conclusions that can be drawn about database design for small scale temperature databases.

A Note on Styles and Symbols

The orthographic form of a word is written in italics (*kvinna*). The meaning of a particular orthographic form is written in single quotes (‘woman’). Semantic concepts and conceptual metaphors are written with capital letters (WOMAN). One-to-many relationships in relational databases are represented by arrows (with the arrowhead pointing to the “many” entity).

Lexical Typology and Temperature Term Research

The last half-century has yielded much on the typology of semantic domains like color, body parts, and kinship systems. The emerging field of lexical semantic typology – sometimes known as semantic typology or lexical typology – focuses on investigating what is language specific and what is universal, in the lexicalization of cross-cultural concepts. While one language might express the semantic difference between two concepts by lexicalization, another language might make use of morphosyntactic tools to encode the difference – for this reason, the lexical semanticist cannot exclusively deal with the lexeme inventories of concepts but must also pay close attention to their morphosyntactic environment.

Temperature terms have emerged as a captivating new area of study in the last decade. Temperature terms directly address not only the physiological reality of temperature, but also the cultural conceptualization of temperature sensations. Following Koptjevskaja-Tamm and Rakhilina (2006):

> Temperature phenomena are universal, relatively easily perceptible by humans and crucial for them, but their conceptualisation involves a complex interplay between external reality, bodily

experience and evaluation of the relevant properties with regard to their functions in the human life. The meanings of temperature terms are, thus, both embodied and perspectival.

There are three main foci in temperature term research:

The first involves lexicalization and categorization and seeks to answer questions such as:

What temperature term concepts are encoded as words across languages?

Are temperature term systems completely free to vary, or are there restrictions?

How can the meanings of temperature terms be described (e.g.) via reference to the objective temperature scale, to the human body and human perception, or to typical entities, like fire of ice?

While English has at least five levels of intensity in its temperature term system (hot, warm, lukewarm, cool, cold), there are several languages that only have two (e.g. Mwotlap) or three (e.g. Yoruba) levels of intensity of temperature terms.

English uses the same temperature terms to refer to most kinds of entities: a room, a bowl, a stone, a person, and water can all be hot, for instance. Several languages restrict the kinds of entities that temperature terms can be used with. In Ewe, for instance, there is a special set of temperature terms that can be used only for water temperatures.

Several languages also have restrictions on which semantic frames temperature terms can be used with. Speakers of languages which acknowledge these semantic frames might treat the temperature of a room (that can be felt with the whole body) as a different concept than temperature that can be felt by touch. Likewise, the internal comfort temperature of humans can be conceptualized as a very different thing than other kinds of temperature perception. These languages might differentiate between the semantic frames by lexical or morphosyntactic means.

Temperature terms are anthropocentric in their use – imagine a cup of tea that is 95 degrees Fahrenheit and a glass of juice that is 95 degrees Fahrenheit. The tea might readily be described as *cold (I didn't drink the tea and now it's cold),* while the juice might be described as *warm* (*the juice is warm now, let's put some ice in it).* Temperature terms are typically not contradictory (as are the adjectives *dead-alive* in English) but contrary. This also means that there are no default antonyms – the antonym to *hot* can sometimes be *cold*, sometimes *cool*. This fact is by no means restricted to temperature but would be applicable to all scalar adjectives.

Another focus area in temperature term research is this interaction between the lexicon and the grammar. Are there cross-linguistic patterns in the kinds of syntactic constructions and word classes that are used for talking about temperature terms? Temperature terms can be adjectives, verbs, and nouns in different languages – temperature information can even be expressed through interjections, like the English *brrrr*! 'It's cold, I'm freezing!'

A third focus area is the semantic extensions that temperature terms can take and the kinds of concepts that are used to form temperature terms. Many languages have temperature terms, such as Estonian *külm* 'cool', that come from temperature bearing objects (in the case of *külm*, from 'frost'). As for the metaphorical and metonymical uses of temperature terms, they vary greatly. Many languages use temperature terms to talk about intensity of emotion. Lakoff & Johnson (1980) postulate a conceptual metaphor INTENSITY OF EMOTION IS HEAT, but some languages use temperature terms only rarely in extended ways or not at all (Vejdemo and Vandewinkel submitted). The typical climate temperature can have influence on the kinds of extended meaning that temperature terms can have according to Al-Haq and El-Sharif (2008): in Arabic, a language typically spoken in quite warm climates, happiness is often described as

'cold' – in colder climates, where European languages are spoken, 'cold' is most often associated with negative emotions[1].

Cross-linguistically, much research remains to be done. Do languages spoken in climatically very hot or very cold regions differ in their temperature term systems – both when they describe the world and when they construct metaphors?

During the last few years Professor Koptjevskaja Tamm has led the research project "Hot and Cold – Universal or Language Specific?" and encouraged researchers all over the world to contribute their data on temperature terms. Many researchers have based their work on temperature on a questionnaire (Koptjevskaja-Tamm 2007), but since this research field is so little explored, the questionnaire is by necessity vague and encourages exploration of the temperature domain rather than answers to particular questions. While the interest generated by the temperature research is heartening, it has also led to a situation where a lot of data about temperature terms is starting to be assembled – data that are disparate in their form and coverage, and from languages that cannot be seen as a representative sample.

Databases – solutions and challenges

This typological research project is a good case study to show examples of the problems facing many small scale efforts in semantic typology. In order to analyze datasets that partly, but not fully, overlap in dozens of languages, a decision was made to construct a database – The Typological Database of Temperature Terms (TDTT).

[1] Though note that this is not true of expressions such as *keep a cool head* – in English *cool* and *cold* are used as vehicles for different kinds of metaphors. The issue of semantically extended uses of temperature terms is complex and largely outside the scope of this paper.

On the surface, databases would seem like an ideal tool for researchers interested in aspects of semantic typology – good databases could be onomasiologically and/or semasiologically structured "nets" of meaning, where any data point can point to any other. Trying to database something as elusive as meaning is quite difficult, however, and a great deal of care has to be taken when undertaking such an enterprise, as shown in the work of Evaert, Musgrave & Dimitriadis (2009). Several databases with a lexical semantic focus, like the monolingual Framenet (Ruppenhofer et al. 2010) and Wordnet (Fellbaum 1999), or the many multilingual EuroWordnets (Rodríguez et al. 1998) and the Multilingual Framenet (Boas 2009), the Kicktionary (Schmidt 2009) or the Catalogue of Semantic Shifts (Zalizniak 2008), have been constructed and published during the last decades. While each posed particular challenges, there are recurring problems.

Examples of Recurring Problems

In this section I intend to discuss some recurring problems that face constructors of small scale lexical semantics research databases. First, there are the twin issues of how to delimit meaning and how to indicate that two lexemes share the same meaning. Another important choice that needs to be made is whether to focus on quantitative or qualitative data. Finally, there is the problem of weighing the opposing needs of complexity and usability in the database design.

The issue of delimiting meaning can be illustrated by research into temperature terms. In Swedish, anything that can be described as *het* 'hot' can also be described as *'varm'*. It is equally correct to say *solen är väldigt het* 'the sun is very hot' as it is to say *solen är väldigt varm* 'the sun is very warm'. *Varm* semantically subsumes *het*. How is this to be expressed in the database – especially considering that in the related language English, *warm* does not subsume *hot,* and this difference needs to be expressed. Another example from the semantic domain of body parts

can be found in Zalizniak (2008): “Consider the Russian word *ruka*. It is an equivalent of both English hand and arm, but Russian explanatory dictionaries do not postulate two different meanings for it, so it is a case of semantic generality.”

Another challenge is the usability of the final database for people with only limited experience with working on computers, versus the complexity of the database structure. In a large scale research project with ample funding for programmers and interface designers, this is not a problem, but most studies in lexical typology are, or start out, small with little or no funding. Not only must the database be easy to use for the users, it must also be easy to maintain even when the person who built it is no longer available and there is little in-house programming knowledge. At the same time, the study of meanings is a very complex field.

Describing differences in meaning between two related lexemes can often result in many pages of text. Keeping this kind of information from qualitative, detailed studies of word meaning, as text entries in the database is an excellent way to provide researchers with information about a few lexemes. The more data that are kept in large chunks of texts, the more similar a database would be to an indexed digital dictionary. This is a viable strategy for building lexical semantics research databases, but it does have several inherent problems. Once there is a desire to compare more than a handful of languages, it is hard to find cross-linguistic patterns in large amounts of text. It is far better to have the textual data encoded as attributes with values from closed sets of possible values. The following example is from research notes on Italian temperature terms (Lami). In the quoted text, we find a lot of interesting data on some Italian temperature terms:

There are temperature terms that have relatively similar meaning and can be considered (quasi-) synonyms to *caldo:* ***calore/tepore*** […] it is used referring to the warmth of a room, a blanket, the fire, the sun, something that protects, in some way; it often has a slight connotation of a positive

warmth and indicates non-tactile temperatures. It is used also in extended meaning to indicate the warmth of a person, of a hug, meaning something familiar and protective: *mi riparai sotto il calore delle coperte*/I took shelter under the warmth of the blankets; *fui avvolto dal calore del suo abbraccio*/I was wrapped by the warmth of his embrace, *bollente* (literally means "something that has reached its boiling point," referring to tactile temperatures, for non-tactile temperatures it evaluates just the wind, and, to a small extent, the air; it has the same meaning of *fa caldo* – it's hot – in form of the verb *bollire*; to boil, at the impersonal form *si bolle*, but it is marked in diatopy as regionalism of Tuscany).

In an attempt to make this qualitative data more quantitatively useful, I structure the information in the following way in a simple table. I attempt to transform the textual data into a (partial) list of features. This way, it is possible to easily see which lexemes can be used with what kinds of entities, but a lot of the fine grained semantic analysis in the text is lost. For instance, the text says that *bollente* us used with wind and <u>to a smaller extent</u> with air. It is not clear how this information is best migrated to a binary (or closed set) semantic feature list.

	Quasi-synonym	**Room**	**Blanket**	**Fire**	**Sun**	**Tactile temperatures**	**Wind**	**Air**
calore, tepore	*caldo*	☑	☑	☑	☑			
bollente	*caldo, fa caldo*					☑	☑	☑

Figure 1. Some semantic features of two Italian temperature terms.

One of the fundamentals of these databases is often a list (either a structured ontology or a "bag," i.e an unordered list) of semantic features that database entries are said to possess. It is by encoding this list that the database gets its structure. Yet the very nature of the research, the

reason that the database was conceived of, was to develop and refine the list. Thus, halfway through the construction of the database, items on the list will get subdivided, or merged, or removed. The database must thus be equipped to handle frequent and fundamental redesigns of its data structure.

Most research into lexical typology involves a great deal of manual analysis of data. This means that databases used in this research must be human readable with good, easily accessible interfaces.

What Should the Ideal Lexical Typology Database Do?

These are the core questions that need to be answered when evaluating the success of small scale lexical typology databases.

- How does the database delimit meaning?
- How does the database express translation equivalences between lexemes?
- Is it easy to implement big changes in the data structure?
- Is the database easy to use for inexperienced users?
- Is the database easy to maintain for inexperienced users?

Often, lexical typology is also interested in the semantically extended meanings of terms, such as metaphors and metonyms. While some projects are specifically interested in a synchronic snapshot of the lexical expressions of a certain semantic domain, many are also interested in diachronic semantic shifts. Certain projects are also specifically interested in the interplay between semantic and morphosyntactic differences. Thus I add the following questions:

- To what degree does the database succeed in capturing extended meanings?
- To what degree does the database succeed in capturing diachronic semantic shifts?

- To what degree does the database succeed in capturing the interplay between morphosyntax and semantics?

Chapter 2: Review of Available Databases

In this section I will review several databases that are used for research into semantic typology. I will first briefly discuss some general database models and languages and then individual databases.

Database Languages and Models

An in-depth discussion of relational, flat file, hierarchical databases or SQL, RDF is beyond the scope of this thesis. This is a brief overview of some terms.

Flat file databases. A list of names and addresses, one person's data per line, is a flat file database. A comma or tab separated value file is another example. A flat file database with an index has an additional file which speeds up searches in the database. This is illustrated below, where lexemes are in one file and another file has information about which language they come from. Please note that this is only one example of how indexes and flat file databases can be constructed.

Relational Databases and SQL

SQL, Structured Query Language, is the most widely used language to manage data in relational database management systems. It is used to add, update, delete, and select data.

File 1:	File 2:
varm, warm, Swedish, gzhaate, warm and sunny, Ojibwe kall, cold, Swedish	Swedish: line 1,3 Ojibwe: line 2

Figure 2. An example of a flat file database with an index, which uses comma separated values.

Relational databases work with tables, which have columns (attributes) and rows (records). All tables have one ID column. Table A is linked to table B by having the ID column of B appear as a column in table A. An example of how a relational database might handle the fact that the words *varm* and *kall* are Swedish and have the English translations 'warm' and 'cold' respectively, while the lexeme *gzhaate* 'hot and sunny' is from Ojibwe, is represented below.

Table A: Lexemes

ID	Lexeme	English Tr	Language
1	varm	warm	1
2	kall	cold	1
3	gzhaate	hot and su	2

Table B: Languages

ID	Language
1	Swedish
2	Ojibwe

Figure 3. An example of a relational database.

RDF databases and XML. RDF stands for Resource Description Framework and is a language used to make statements about resources. Resources can exist in the real world, or be pieces of

Subject	**Predicate**	**Object**
Lexeme1	is a	Lexeme
Lexeme1	has written form	Varm
Lexeme1	is from the language	Swedish
Lexeme1	has the English translation	Warm
Lexeme1	has the researcher comment	Look up this word in that green book
Lexeme2	is a	Lexeme
Lexeme2	has written form	Kall
Lexeme2	is from the languages	Swedish
Lexeme2	has the English translation	Cold
Language1	is a	Language
Language1	has language name	Swedish

Figure 4. An example of some RDF triples.

information or concepts. RDF triples are made up by a subject, a predicate and an object – these are called elements. RDF can be queried by the SPARQL language, but SPARQL can only select data, not add, delete or manage it. RDF is a W3C recommendation[2].

RDF is intended for situations in which information needs to be primarily processed by applications, rather than humans. It makes use of URIs (Uniform Resource Identifiers) as unique IDs for most subjects, predicates, and objects – though a subject or an object can also be a string literal. In Figure 4, all subjects, predicates, and objects could be identified using unique URIs except probably the forth object "Look up this word in that green book." The other elements, like "varm" or "is a" or "lexeme," will most likely be used several more times, but the object "Look up this word in that green book" is most likely a onetime occurrence.

RDF Schemas are sets of rules about, for example, which kinds of objects predicates can take. One such rule might be that the predicate "is from the language" can only take objects that themselves are the subjects in a triple with the predicate "is a" and the object "language."

Many RDF documents make use of the Dublin Core[3] set of elements (defined by URIs), which are designed to describe documents and other written resources.

There are several different syntaxes defined to express RDF statements. One of the most popular strategies is to use XML, Extensible Markup Language. XML is a W3C recommendation.[4] It specifies a way of structuring information in a standardized fashion. An XML document consists of elements represented by tags <…>. Elements can be named anything, but an element has to

---

[2] See http://www.w3.org/TR/rdf-primer/

[3] See http://dublincore.org/

[4] See http://www.w3.org/TR/xml/

have a start tag(<tag>) and an end tag (</tag>). If a tag is written in the document after a tag <a>, the end tag of () has to precede the end tag of <a> ().

The EXtensible Stylesheet Language (XSL) and EXtensible Stylesheet Language Transformations (XSLT) are additional languages used to select, modify, add, and delete data from XML documents.

XML can be used for other databases than RDF databases. An author can define his or her own set of rules (in an XML schema or a Data Type Definition document) that govern how the XML documents may be structured.

Databases Used for Lexical Semantics Research

Some of the databases I will discuss in this section are restricted to a single semantic domain, (e.g. The Kicktionary), while others (e.g. the Multilingual FrameNet), deal with a great variety of lexical expressions. To understand the structure and motivation for the multilingual FrameNet and the Kicktionary, I must also discuss their inspiration, the monolingual Framenet – and the same is true of the monolingual Wordnet, which was the inspiration to EuroWordnet. Framenet, Kicktionary, and Multilingual Framenet are related – the latter two are based on the first and they all operate within Frame Semantics.

While the original Framenet and the Multilingual Framenet are intended to be used by both humans and computers, the Kicktionary is primarily envisioned as a resource for humans.

The EuroWordnet project was constructed on the work already done on the Wordnet database. Both of these databases are intended to be used for NLP research as well as for more traditional semantic research.

With the exception of the Catalogue of Semantic Shifts, the databases are all available to the public. This means that there are many different versions of them – for instance, FrameNet was originally constructed with SGML markup of texts and was later put both into XML and into a relational MySQL database. Other researchers have described methods for transforming this relational database into an RDF database (Narayanan et al. 2003). Similarly, EuroWordnet was originally constructed as a flat file database with an index and was accessed through the software Polaris (Vossen 1998). De Luca and Nürnberger converted EuroWordnet to XML (2006) and later to RDF (De Luca, Eul, and Nürnberger 2007).

Following is a short summary of the databases I intend to review.

Table 1. Overview of some databases often used in lexical semantics research.

Name	**Data**	**Research Methodology**	**Data Structure**
Framenet	Monolingual, general dictionary entries	Frame Semantics	SGML, XML, Relational
Kicktionary	Multilingual soccer terms	Frame Semantics	XML
Wordnet	Monolingual, general dictionary entries	NLP and general semantic research	text files, SGML, XML, RDF
EuroWordnet	Multilingual general dictionary entries	NLP and general semantic research	text files, SGML, XML, RDF
Catalogue of Semantic Shifts	Nouns	Diachronic semantics	SQL
Hamburg Metaphor Database	Selected metaphors in newspaper texts	Conceptual metaphor studies	SQL, RDF/OWL

The Catalogue of Semantic Shifts

Motivation

A semantic shift, as defined by Zalizniak (2008), is when the meaning of a lexical unit changes. In French, the lexeme *femme*, which at one point only meant ‘woman, now also means ‘wife’. A semantic parallel is when a semantic shift can be found in many languages; for instance, German *Frau* underwent the same change. Similarly, in Russian, the lexeme *zena* means ‘wife’ but has lost its original meaning of ‘woman’. Finding semantic parallels is important for etymological research; if a change from one concept to another is already known from a language, it is more likely to appear in others.

The Catalogue of Semantic Shifts was created to store large amounts of data on semantic shifts and to enable researchers to look into cross-linguistic patterns.

So, the Catalogue of Semantic Shifts might be used: (1) as a semantic plausibility criterion in linguistic reconstruction; (2) as a basis for semantic typology; (3) as a linguistic evidence for the nature of cognitive processes; (4) as a contribution to the history of ideas (Zalizniak 2008, 219).

In addition to this, Zalizniak (2008) says that the database will help to identify areal tendencies and to trace the ways of cultural influences.

Origin and extent of data. The database includes semantic shifts for which at least two independent realizations can be found. Most data come from cases of synchronic polysemy, not from etymological reconstruction. A wide variety of data sources have been used, including dictionaries and scholarly works describing semantic shifts within particular domains. The authors also did corpus studies and field work to verify some of the data. The database contains

(at least, since some time has passed since the article was published) 3000 semantic shifts, with more than 8700 realizations in 319 languages (Zalizniak 2008).

Database structure. The catalogue consists of ten tables, as can be seen in Table 2. The tables are linked through foreign keys, as when the attribute source in the table Catalogue has a reference to an entry in the table Meanings (see Table 3 below). Each row in the table Catalogue encodes a particular concept as a source domain and another concept as a target domain and can be referenced in turn by an entry in the table Realizations. Thus, a row in the table Catalogue can specify that the meaning WOMAN can turn into the meaning WIFE and two separate entries in the table Realizations (an overview of this table can be seen in Table 3) could be linked to this through a foreign key (in the attribute ID_Shift) to illustrate that French femme and German Frau underwent this change.

Table 2. All tables in The Catalogue of Semantic Shifts (adapted from Zalizniak 2008).

Table name	Storing
Catalogue	Information about semantic shifts.
Related_Shifts	A list of semantic shifts that are in some respect similar to a particular semantic shift.
Realizations	Information about realizations of the semantic shifts.
Types of Realization	A list of possible relation types between two lexical items within a realization.
Languages	A list of languages.
Meanings	A list of meanings that are used as source or target meanings in the database.
Taxons	An ontology used to classify meanings.
Contributors	Names of contributors to the database.
Status	A list of possible current statuses of a semantic shift.
References	A list of references.

Table 3. The Catalogue table in The Catalogue of Semantic Shifts (adapted from Zalizniak 2008).

Table name	Storing
ID	ID number
Source	Source meaning, links to table: Meanings.
Target	Target meaning, lnks to table: Meanings
Direction	Direction of the semantic shift (forward, backward, mutual).
Status	Current status of the shift: accepted, new, suspended, not accepted, links to table: Status
Comments	Comments on the semantic shift
Contributor	Name of the person who contributed te semantic shift, links to table: Contributors.

The concepts WIFE and WOMAN each have an entry in the table Meanings.

Table 4. The Realizations table in The Catalogue of Semantic Shifts (adapted from Zalizniak 2008).

Column	Storing
ID	ID number
ID_Shift	ID number of the semantic shift being documented, links to other table: Catalogue.
Language_Source	Language having the lexeme with the source meaning.
Lexeme_Source	Lexical item demonstrating the source meaning.
Lexeme_Source_Translation	Translation of the Lexeme_Source.
Example_Source	Context showing the use of the Lexeme_Source.
Language_Target	Language having the lexeme with the target meaning.
Lexeme_Target	Lexical item demonstrating the target meaning.
Lexeme_Target_Translation	Translation of the Lexeme_Target.
Example_Target	Context showing the use of the Lexeme_Target.
Realization_Type	Type of realization, links to another table Types of Realization.
Direction	Direction of the shift (forward, backward, mutual).
Comments	Comments.
Status	Current status of the realization.
Contributor	Name of a person who ontributed the realization, links to other table: Contributors.

Interface and implementation. The Catalogue is at present not available online. The interface is available in both English and Russian. It is built on a Microsoft Access 2002 platform with several different forms and reports made available to users. There is no web accessible interface and any changes must be done in a single master database.

Challenges and solutions. The catalogue's main focus is not on lexemes but on semantic concepts in general and in particular on how these concepts are involved in semantic shifts. Apart from semantic changes in lexemes, it also focuses on grammaticalizations.

Two lexemes or phrases (realizations) from different languages are considered semantically close when they are connected to the same semantic shift. Differences in the way these individual realizations reflect the underlying semantic shift are captured through text comment fields. There is a danger that small important differences in meanings between different realizations are overlooked. As Zalizniak (2008) points out:

Should we consider, for example, the shifts from source meanings 'to catch', 'to grasp', 'to reach' or 'to get' to the target meaning 'to understand' as the one and the same or as four different shifts? Or should pairs of meanings 'to see' ~ 'to look at' be treated as separate labels (which participate in semantic shifts independently), or as conjoined single labels like 'to see/to look at'? On the one hand, if we unite such "close" meanings we risk losing the substantial difference between them that can produce different metaphoric development and therefore different semantic shifts (cf. 'to see' _ 'to know' vs. 'to look at' _ 'to care for'). On the other hand, there are languages where these meanings are expressed syncretistically, and for these languages there is no way to distinguish between two source meanings in question (e.g. between 'to see' and 'to look at' etc.).

The use of Microsoft Access as DBMS means that users have access to a modified SQL query language and that they can easily, through SQL queries or graphic interaction with the data, implement changes in the database. There are many published books, websites and tutorials on the inner workings of the Microsoft Access DBMS, which means that even inexperienced users can interact with the database in a short amount of time. The basic forms and reports that the

database generates can be built and modified with little or no prior knowledge of programming, though more advanced forms that might be displaying data from many tables at once, might require some knowledge of the Visual Basic programming language.

The catalogue is not specifically intended to address questions about extended semantic meanings. Indeed, from a diachronic point of view it is not clear what an extended meaning is.

In the table **Realizations**, there is a text field attribute, **Example_Target**, where a user can input an example. There is unfortunately no way to store information about, for instance, differences in attributive or predicative uses of the lexeme realization.

FrameNet

Motivation. Frame Semantics was first introduced by Fillmore (1992), one of the key figures behind the publicly available Frame Net Database housed at http://framenet.icsi.berkeley.edu/ (Ruppenhofer et al. 2010).

Core Frame Elements:

Body_part: A part of the body of the Experiencer in which the sensation of a temperature is located.

"My hands are FREEZING."

Experiencer: The entity which perceives a level of warmth.

Non-Core Frame Elements:
Degree: How much the actual feeling of temperature deviates from a certain level of warmth.
Definition:

An Experiencer senses different degrees of warmth that may or may not be related to the ambient temperature. The level of warmth is usually compared against the Experiencer's subjective standard of comfort.

My feet are COLD.
The fire will keep you WARM.
I'm too HOT with this sweater on.

Figure 5. The Subjective Temperature Frame.

A basic idea in Frame Semantics is that the meaning of a lexeme cannot be fully understood until all its semantic and syntactic valences have been mapped. A Semantic Frame is a "script-like conceptual structure that describes a particular type of situation, object, or event and the participants and props involved in it" (Ruppenhofer et al. 2010). Petruck (1996, 1) has the following definition: "A FRAME is any system of concepts related in such a way that to understand any one concept it is necessary to understand the entire system; introducing any one concept results in all of them becoming available."

Each frame has a number of Frame Elements – a kind of semantic roles. For instance, in FrameNet, the Subjective_Temperature-Frame has the Frame Elements BODY-PARTS and EXPERIENCER as Core Frames and DEGREE as a Non-Core Frame. As shown in Figure 5, in the example sentence *I'm too hot, I* is an Experiencer FE and *too* is a Degree FE.

There are several Lexical Units connected to each frame. The LUs *burn up.v, freezing.a, cold.a, cool.a, hot.a, warm.a* are connected to the Subjective_Temperature frame. Each lexical unit has a word class, identified with a period and a letter (a for adjective, v for verb etc.) after the name of the LU. Each of the LUs has a lexical entry with a definition and information on which FEs are used with it and in which syntactical positions they occur.

When a lexeme is polysemous, the different meanings are identified as belonging to different Semantic Frames. Viberg (2001) says that "Framenet [...] analyses the argument structure of verbs and nominalizations, based on frame elements, which are a kind of domain specific deep case roles." An example is the FE *warm*, which appears in the following frames[5]:

[5] See at http://framenet.icsi.berkeley.edu, accessed 2010-11-01

warm.a (Ambient_temperature frame)

warm.a (Social_interaction_evaluation frame)

warm.a (Temperature frame)

warm.a (Color_qualities frame)

warm.a (Risky_situation frame)

warm.a (Subjective_temperature frame)

warm.a (Heat_potential frame)

warm.v (Cause_temperature_change frame)

warm.v (Inchoative_change_of_temperature frame)

The letter after the period (a or v) is the word class: adjective or verb. The definition of *warm.a (Subjective_temperature frame)* is "comfortable in terms of body heat," while the definition of *warm.a (Ambient_temperature frame)* is "having an environment with a fairly or comfortably high temperature." The definition of *warm.a (Risky_situation frame)* is "uncomfortable due to potential danger" – but there is nothing identifying *warm.a (Risky_situation frame)* as a semantically extended use of the concrete *warm*.

Origin and extent of data. The Framenet data have been entered manually and has lexemes from a wide variety of word classes. Once work on a particular frame has been started, it is desirable to finish entries for all lexemes that belong to the frame.

Database structure. There have been several instantiations of the FrameNet Database. The first Framenet, referred to as FrameNet I in Baker et al. (2003) was constructed in SGML, a markup language that preceded XML. The example below shows a bit of SGML markup of the annotated sentence "I am conscious that it is a complex and difficult question." Inside the <C> tags, certain parts of the utterance are given tags in attributes (the nature of which we shall not discuss at the moment).

<S TPOS = "81597120">…

<C FE = “Cog” PT = “NP” GF = “Ext”>I </C>
<C TYPE = “SuppV” PT = “XFE” GF = “XFE”>am </C>
<C TARGET = “y”>conscious </C>
<C FE = “Cont” PT = “Sfin” GF = “Comp”>that it is a complex and difficult subject</C>
</S>

The next instantiation of FrameNet, FrameNet II, was moved to a MySQL database. The reason for this was twofold: the need for an easy way to handle many-to-one or many-to-many relationships and the problem with repeated information (if the name of a frame element needed changing, it would have to be done in potentially thousands of places in the SGML markup) (Baker, Fillmore, and Cronin 2003).

The MySQL version of FrameNet has three interconnected databases, according to Baker, Fillmore, and Cronin (2003). "The Lexical Database contains the relationships of word forms, lexemes, lemmas, and their parts of speech. The Frame Database defines and interconnects frames and their FEs. The Annotation Database contains annotations and sentences, which comprise the majority of the FrameNet data.”

Baker, Fillmore, and Cronin (2003) discuss the possibility of creating a hybrid database solution for FrameNet:

"In the longer term, in a third phase of the project, we might consider using a different sort of database to handle information about semantic types and frame-to-frame relations. These tables typically contain relatively few entries, each entered “by hand,” so there is not much need for the speed provided by automatic indexing of tables in the relational database. Also, changes in our definitions of relations often require substantial changes in the structure of the relational database and the software which accesses it. For these reasons, it may be advantageous to use a hybrid approach, with a relational database for the ‘heavy lifting’ required in the annotation process,

accessing the lexemes, etc., and a more flexible database (such as lisp or prolog clauses) to handle the hand tailoring of relations among frames, FEs, semantic types, etc."

Other researchers have described methods for transforming this relational database into an RDF database (Narayanan et al. 2003).

Interface and implementation. Framenet has several interfaces available online[6] - they require no prior knowledge to use. It is possible to interact with the database through HTML pages (rendered from the XML source and XSL style sheets) focusing on either frames or lexical units or to query the database through the FrameSQL website[7]. The latter queries a MySQL relational database version of Framenet but does not allow for text based SQL queries. It does offer a graphic interface where a user can build a query using drop-down menus and lists. It is also possible to submit a request to download Framenet in XML format.

Challenges and solutions. The Frame Semantic approach to semantic content is that the meaning of a lexeme can only be grasped by understanding the context is appears in. Each time a lexeme appears in a frame, it is treated as a different unit than the same lexeme in another frame. This means that no attempt is made to differentiate homonyms and polysemes: that is, there is no place to store information about the fact that warm.a(Ambient_temperature) and warm.a(Subjective_temperature) are semantically closer to one another than chest.n(observable_body_parts) and chest.n(containers).

---

[6] See http://framenet.icsi.berkeley.edu/

[7] See http://sato.fm.senshu-u.ac.jp/frameSQL/fn2_15/notes/

The Framenet interfaces have been produced, and are maintained, through research grants that have allowed for significant investment in programmer time, and the interest that Framenet has generated has also led to the data being presented in several different formats, such as RDF (Narayanan et al. 2003).

Of the lexical typology databases that are readily available for researchers, Framenet (and its successors Multilingual Framenet, Kicktionary etc.) are the resources that most clearly focus on Morphosyntax. The use of a certain lexical unit as a frame element in a particular frame is always exemplified by carefully tagged example sentences.

Framenet was never supposed to contain any diachronic data and there is no system in place to store such information in the database.

The Kicktionary

There have been many research projects focusing on creating multilingual Framenets over the years. Many are focused on NLP implementations and have lexemes from many different word classes (Boas 2009). The Kicktionary database is domain restricted to footballs terms and is primarily built for humans, not programs.

Motivation. The Kicktionary is a multilingual (English, French, and German) domain specific (it is restricted to soccer terms) frame semantic database. Its primary goal is to be a resource for humans, it is not specifically intended to be harvested by computers for purposes of machine translation or NLP research (Schmidt 2009).

By building a domain specific database, constructed on frame semantic principles, Schmidt (2009) hopes to investigate which advantages such a database would have over other printed or electronic lexical resources; how corpus examples and multi-medial elements (such as

illustrative images), can be successfully integrated with such a database; and if and how the frame semantic principles support the creation of multilingual and domain specific lexical resources.

Origin and extent of data. The Kicktionary was constructed by manual tagging of a corpus of soccer match reports taken from the UEFA webpage (Union of European Football Association[8]). The Kicktionary contains nouns, verbs and adjectives that are used in this corpus and that fill the role of frame elements in the frame semantic scenes envisioned by the Kicktionary creators (Schmidt 2009).

Typically, when multilingual Framenet databases are created, the English content is removed, but the data structure (such as the frames) stay. The new database is repopulated with data from other languages, with appropriate changes to the data structure where such are needed. In the Kicktionary, the data structure was populated with English, German, and French at the same time. All in all, it deals with approximately 2000 soccer terms (Schmidt 2009).

Like Framenet, The Kicktionary has frames. But in The Kicktionary, frames are grouped together into larger units, called scenes. Such a scene is Goal, which has the frames **Goal**, **Overcome_Goalkeeper, Concede_Goal, Own_Goal, Prepare_Goal, Convert_Chance, Award_Goal, Celebrate_Goal, Multiple_Goals, Score_Goal**, and **Start_End_Match**. Both scenes and frames are explained by text, as well as illustrative graphs and photos.

The frame **Celebrate_Goal** has the following frame elements:

[8] http://www.uefa.com

1) Goal

2) Scorer

3) Team_Mate

4) Scorer_Team

Several lexical units are also associated with the frame **Celebrate_Goal**. The LU *bejubeln* and *feiern* are from German. There is only one English LU, *celebrate*. *célébrer* and *fêter* are French. It is possible to detect small differences between these lexical unit, by looking at which frame elements they can be associated with. In figure X, the numbers on top of the columns represent the four frame elements mentioned above. German *bejubeln.v* involves all four frame elements, while *feiern.v* involves only the first three. While *bejubeln.v* is closely related to French *célébrer.v*, we can see in Table 5 that German *feiern.v* and French *célébrer.v* are in fact much closer, semantically speaking.

Table 5. Lexical unites in the Celebrate_Goal frame.

*	1	2	3	4
bejubeln.v	X	X	X	X
feiern.v	X	X	X	
celebrate.v	X	X		X
célébrer.v	X	X	X	
fêter.v	X	X	X	

Lexical units are grouped into synsets. These synsets contain words from multiple languages – the five lexical units in Table 5 are one such synset. Synsets can have lexical relations (hyponyms/hypernyms and holonyms/meronyms for nouns and troponyms for verbs) with other synsets, but it is not mandatory.

Database structure. The Kicktionary is stored in several XML files. One file contains all the lexical unites as well as their annotated examples, information about their assignments to a certain frame and to a certain synset.

```
<EXAMPLE lang="en" source-element-id="p5" source-text-id="75236">
        <FE_REF fe-idref="LEADER">Latvia's</FE_REF>
        <LU_REF lu-idref="lead.n">lead</LU_REF>
        <txt>, however, was short-lived.</txt>
</EXAMPLE>
```

Figure 6. Excerpt from the XML file that stores the tagged examples from the Kicktionary.

In Figure 6 the Frame Element *Latvia* is labeled with the ID "LEADER" and the lexical unit *lead* is labeled with the ID "lead.n."

Another file contains the concept hierarchies, i.e information about the lexical relations that can exist between synsets.

Each of the 16 scenes that comprise The Kicktionary has its own file, which describes the scene in general and which has information about which frames belong to the scene.

Interface and implementation. The Kicktionary can be accessed through the project web page[9]. The data are stored in XML format and HTML pages are generated by an XSL style sheet. (The XML data files can be requested by contacting the web master.)

[9] http://kicktionary.de/

The online interface lets users interact with the database starting either with lexical units, frames or a taxonomy of lexical relations.

Challenges and solutions. It is important to note that The Kicktionary never explicitly states that two lexical units from different languages have or do not have the same semantic meaning. Rather, their semantic closeness must be inferred from the number of frame elements they have in common. To repeat an earlier example, the German lexical unit *feiern.v* and the French *célébrer.v* can be inferred to be semantically closer than the French lexical unit is to the German *bejubeln*. The reason for this is that the former two share three frame elements in the frame Celebrate_Goal (namely they both involve the frame elements Goal, Scorer and Team_Mate), while *bejubeln.v* also has a fourth frame element (Scorer_Team). To find further evidence of similarity or difference between *feiern.v* and *célébrer.v*, the researcher must read their descriptions.

While this approach might be too vague if the intended user were a computer program making semantic inferences from the database, it is eminently suitable for human readers – the stated user group of The Kicktionary. It is possible to find the semantic data categorized just enough to be able to spot interesting cross-linguistic patterns, but the last step in examining the semantic similarities between lexical units is left for the researchers.

The fact that each lexical unit can only belong to a single frame is a problem in the current implementation of the database. Schmidt (2009) acknowledges this problem and gives the example that the LU kick-off belongs to the Match scene, but it could just as easily have been placed within the Goal scene, since there is a kick-off after every goal.

There is no linguistic reason that a particular lexical unit must be restricted to a single frame. I believe that if the relationship between a lexical unit and a frame was stated in the XML file

containing the frame, rather than as a tag in the XML file describing the lexical unit, each lexical unit could be associated with a many frames and each frame could be associated with many lexical units.

The choice to keep The Kicktionary in XML files means that the data are eminently compatible with most database software, should its author desire to use it for another purpose than to generate The Kicktionary web page. The most obvious problem with the XML format will be apparent when someone wants to make a database wide change, like changing the name of a particular scene or frame. As long as the XML data conform to its schema, this can be handled through XSLT update scripts. There is thus a need to validate the XML database against the schema whenever a change is made.

It can be noted that each example sentence can only be used to illustrate a particular frame, never more. The Kicktionary provides some parallel annotated texts, like the following example from English (Schmidt 2009):

"FC Barcelona became the first visiting[frame:Away_Game] team[Frame:Team] to win[Frame:Victory] a UEFA Champions League match [Frame:Match] at Celtic FC as goals[Frame: Goal] from Deco, Ludovic Giuly and the homecoming Henrik Larsson secured maximum points in their Group F opener."

The online version of this text lets users click on the lexical units representing each frame (underlined in the text above). They will then be shown information about the frame in question, complete with several example sentences. There are five different frames that could be illustrated by this sentence (**Away_Game, Team, Victory, Match** and **Goal**), but the sentence is only used as an example for the **Match** frame.

This is not necessarily a problem, since the goal of The Kictionary project is not to correctly annotate texts, but to use texts as examples of the frames. It does mean, however, that example sentences in which many frames interact are presented to the user with only one frame. I believe that show casing examples where frames interact could be more illustrative.

Wordnet

Wordnet is one of the best known lexical databases. It is monolingual and I address it here mainly as a precursor to one of its multilingual successors, the EuroWordnet. It was manually assembled and has been used in many subsequent research projects. Because Wordnet was initially conceived as a test bed for a particular model of lexical organization that had never before been implemented on a large scale, it had to be manually constructed. Good implementations of semantic networks did not exist before Wordnet and researcher did not know which kinds of relations would be necessary to create a network linking the bulk of the English lexicon (Fellbaum 1999).

Motivation. The goal of Wordnet is to be a computer harvestable online dictionary that can be used for computational research into semantics. It also has several online interfaces researchers can use to look up particular words and their semantic relations.

Origin and extent of data. Wordnet contains compounds, phrasal verbs, collocations and idiomatic phrases, but the word is the basic unit. Wordnet does not contain organizational units larger than words, such as the frames do in Framenet (Fellbaum 1999).

Wordnet's strict separation of its lexical data into syntactic categories (nouns, verbs, adjectives and adverbs) precludes the data being used for framelike semantics. Instead, Wordnet makes use of relational semantics between words to relate words from a common semantic domain.

Wordnet is divided into four separate semantic nets, one for each of the word classes nouns, verbs, adjectives and adverbs.

The basic building block of Wordnet is the synset. A synset consists of all the words that express a semantic concept. Synsets are linked to each other by means of a number of relations (hyponymy, meronymy, entailment etc.).

Initially, Wordnet's synsets were intended to contain only pointers to other synsets, but it was found that definitions and illustrative sentences were needed to distinguish closely related synsets whose members were polysemous.

There are two different kinds of relations in Wordnet: relations between concepts and relations between words. Wordnet clearly distinguishes the lexical and conceptual level. Lexical relations hold between semantically related word forms; semantic relations hold between word meanings. These relations include (but are not limited to) hypernymy/hyponymy (superordinate/subordinate), antonymy, entailment, and meronymy/holonymy.

The meaning of a lexeme in Wordnet is thus encoded in a short explanatory sentence and can also be inferred (if it's an adjective) from its antonyms and which other synsets are tagged as "similar to" adjective's synset. If the lexeme is a noun, its synset will have several semantic relations with other synsets (such as hyponomy etc.). Verb synsets will also have semantic relations with other synsets and might have information about "sentence frame" – detailing the default morphosyntactic behavior of the verbs in the synset.

Wordnet marks contradictory adjective pairs (dead-alive) but does not mark contrary adjective pairs (hot-cold). (One of the opposites in a contradictory antonym has to be true, but that is not

the case for contrary antonyms – or in other words, opposing gradable adjectives are contrary antonyms).

Wordnet's structure is primarily mapped onto the lexicon of English, but not wholly. It contains concepts that are not lexicalized in English, but are lexicalized in other languages. One example of this is the concept WHEELED_VEHICLE, which is not lexicalized in English: but if this concept does not exist in the database, there is a problem that *wheel* is a meronym of *vehicle* and all hyponyms of *vehicle* should inherit its meronyms – yet this would mean that *sled* cannot be characterized as a vehicle (Fellbaum 1999).

Database structure. There are several different versions of the Wordnet database. Researchers have developed local interfaces to the publicly available data using .NET/C#, dBase, Java, MySQL, Perl, PHP, PostgreSQL, Python, Ruby, SQL, Visual Prolog/Prolog and XML., among other languages[10]. The Princeton Wordnet[11], still uses a flat file with index database model.

The Wordnet system has four parts: lexical source files, the software used to convert these files into the lexical database, the Wordnet database itself and the software tools used to access the database.

In the lexical source files, lexicographers enter synsets, using a specific syntax developed just for Wordnet. This syntax includes special meaning given to symbols such as =>. If two homonyms are entered (*old* 'not young' and *old* 'not new), they are distinguished by letters being added to their orthographic representation – in the same way homonyms are distinguished in printed

[10] See http://wordnet.princeton.edu/wordnet/related-projects/ for an up to date list.

[11] http://wordnet.princeton.edu/

dictionaries. Each word in a synset is represented by its orthographic word form, its syntactic category, its semantic field (lexical file name) and unique ID number. Together these items make a "sense key" that identifies the word (Fellbaum 1999). The choice to give meaning (not documented in the data files themselves) to symbols such as an equal character = and a more than character > and to distinguish homonyms by letters added to their orthographic form would have been fine if the primary users of Wordnet had been humans. But since the main intended "demographic" for Wordnet are computer programs, this created problems and has led to the data being reanalyzed using several different data models[12].

Excerpt from Wordnet index file:

red-hot a 5 1 & 5 0 02209820 01701218 01304989 01297910 01017083

Excerpt from Wordnet data file:

02209820 00 s 05 juicy 0 luscious 0 red-hot 0 toothsome 0 voluptuous 0 002 & 02207958 a 0000 + 05149234 n 0502 | having strong sexual appeal; "juicy barmaids"; "a red-hot mama"; "a voluptuous woman"; "a toothsome blonde in a tight dress"

01701218 00 s 02 hot 0 red-hot 0 001 & 01700277 a 0000 | newest or most recent; "news hot off the press"; "red-hot information"

01304989 00 s 02 red-hot 2 sizzling 2 001 & 01304348 a 0000 | characterized by intense emotion or interest or excitement; "a red-hot speech"; "sizzling political issues"

01297910 00 s 01 red-hot 0 001 & 01295235 a 0000 | glowing red with heat

01017083 00 s 03 blistering 0 hot 0 red-hot 0 001 & 01015998 a 0000 | very fast; capable of quick response and great speed; "a hot sports car"; "a blistering pace"; "got off to a hot start"; "in hot pursuit"; "a red-hot line drive"

01297985 00 s 01 scorching 1 001 & 01295235 a 0000 | hot and dry enough to burn or parch a surface; "scorching heat"

Figure 7. Excerpt from Wordnet data and index files.

[12] See http://wordnet.princeton.edu/wordnet/related-projects/ for an up to date list.

The Wordnet database can be reconstructed at any time from the lexical source files. The source files are kept in a simple ASCII format. The information for each synset begins at a specific byte offset ("synset address") in a data file and continues until a newline character is reached. This synset address is saved in an index file and the data are saved in a data file. A search for all senses of a word involves doing a binary search for the base form of the word in the index file for a syntactic category, and moving down the list of synset addresses. For each address found, the corresponding synset is read from the corresponding data file (Fellbaum 1999).

Each line (excluding the header) in an index file starts with a word form, a count of the number of polysemous uses the word form has, a list of symbols for all pointers used in the synsets containing the word, a list of synset addresses (one for each sense of the word) (Fellbaum 1999).

In Figure 7, I show an excerpt from the Wordnet index and data fields. The index file has a line for the orthographic form *red-hot*, which as five synsets (the number listed after the orthographic form). It also has five ID numbers listed. These ID numbers can be found in the data file, with more information on these particular uses of *red-hot*. There is nothing linking *red-hot* to *scorching*, however (ID: 01297985).

Interface and implementation. There have been many different interfaces constructed for Wordnet over the years – a current list can be found here:

http://wordnet.princeton.edu/wordnet/related-projects/

The original Wordnet can be accessed through a Princeton website[13]. The interface is rather sparse but very informative. A search for *cool* gives 11 returns: two nouns, three verbs and eight adjectives. Two of the adjectives are presented as follows (Princeton Wordnet 2010):

- S: (adj) cool (neither warm nor very cold; giving relief from heat) "a cool autumn day"; "a cool room"; "cool summer dresses"; "cool drinks"; "a cool breeze"
- see also
    - S: (adj) cold (having a low or inadequate temperature or feeling a sensation of coldness or having been made cold by e.g. ice or refrigeration) "a cold climate"; "a cold room"; "dinner has gotten cold"; "cold fingers"; "if you are cold, turn up the heat"; "a cold beer"
- similar to
    - S: (adj) air-conditioned (cooled by air conditioning)
    - S: (adj) air-cooled (cooled by a flow of air) "an air-cooled engine"
    - S: (adj) caller (providing coolness) "a cooling breeze"; "`caller' is a Scottish term as in `a caller breeze'"
    - S: (adj) precooled (cooled in advance)
    - S: (adj) water-cooled (kept cool or designed to be kept cool by means of water especially circulating water) "a water-cooled engine"
- attribute
    - S: (n) temperature (the degree of hotness or coldness of a body or environment (corresponding to its molecular activity))
- antonym
    - W: (adj) warm [Opposed to: cool] (having or producing a comfortable and agreeable degree of heat or imparting or maintaining heat) "a warm body"; "a warm room"; "a warm climate"; "a warm coat"
- derivationally related form
    - W: (n) coolness [Related to: cool] (the property of being moderately cold) "the chilliness of early morning"

- (adj) cool (fashionable and attractive at the time; often skilled or socially adept) "he's a cool dude"; "that's cool"; "Mary's dress is really cool"; "it's not cool to arrive at a party too early"
- domain usage
    - S: (n) colloquialism (a colloquial expression; characteristic of spoken or written communication that seeks to imitate informal speech)
- similar to
    - S: (adj) fashionable, stylish (being or in accordance with current social fashions) "fashionable clothing"; "the fashionable side of town"; "a fashionable cafe"

[13] http://wordnetweb.princeton.edu/perl/webwn

- derivationally related form
    - W: (n) coolness [Related to: cool] (calm and unruffled self-assurance) "he performed with all the coolness of a veteran"
- antonym
    - W: (adj) unfashionable [Indirect via fashionable] (not in accord with or not following current fashion) "unfashionable clothes"; "melodrama of a now unfashionable kind"

Like many synsets in Wordnet, the two synsets above only contain one single lexeme. Synsets can also contain several lexemes that are synonyms.

The first adjective is a concrete temperature term, the second a metaphor. There is nothing that links the metaphorical use of the lexeme to its source domain.

There are direct and indirect antonyms in Wordnet. The first adjective synset above has a direct antonym: *warm*. The latter example has an indirect antonym via *fashionable*. The reason for this is that the adjective defined as *cool (fashionable and attractive at the time; often skilled or socially adept)* is coded as being "similar to" the adjective *fashionable*. Adjectives that are "similar to" each other inherit each other's direct antonyms as indirect antonyms.

Challenges and solutions. The fact that Wordnet marks contradictory adjective pairs (dead-alive) but does not mark contrary adjective pairs (hot-cold) is problematic for, for instance, temperature term research. In general, Wordnet is best used for large scale semantic research than for detailed semantic research into specific semantic domains. There is little information on which lexemes are metaphorical and there is no diachronic data.

EuroWordnet

Motivation. EuroWordnet is a multilingual lexical database inspired by Wordnet. Roughly speaking, it consists of several monolingual "wordnets," interlinked through an Inter Language Index (Rodríguez et al. 1998).

Origin and extent of data. EuroWordnet has data from Dutch, Spanish, Italian, English, French, German, Czech and Estonian. It is not built on a data stripped version of the English Wordnet – this would have created a heavy bias in favor of English favored types of constructions. Instead, all the wordnets are built independently, following a set of agreed upon rules (Rodríguez et al. 1998).

All wordnets share the same Top Ontology and Domain Hierarchy. The Top Ontology is a semantic ontology, a hierarchy of language-independent concepts, reflecting important semantic distinctions, e.g. Object and Substance, Dynamic and Static. The Domain Hierarchy contains knowledge structures grouping meanings in terms of topics or scripts, e.g. Traffic, Road-Traffic, Air-Traffic, Sports, Hospital, and Restaurant. (It is the closest to Frame Semantics EuroWordnet comes) (Rodríguez et al. 1998).

The vocabulary to be included in EuroWordnet has to be generic: all general word meanings on which more specific concepts depend must be included and it should also store those meanings that are used most frequently. The different wordnets should store roughly the same kinds of semantic concepts, but the language specific lexicalization patterns must still be reflected (Rodríguez et al. 1998).

Database structure. To ensure that the individual characteristics of languages were kept at the same time that all the wordnets covered approximately the same semantic areas, a first version of

the different wordnets were built independently (and fitted into a Wordnet inspired synset structure) and then compared. From this, a common set of synsets that had at least two of the individual wordnets were chosen. All wordnet creators had to find lexical data representing these synsets. After data for these new synsets were gathered in the wordnets, they could again be compared. The common core of synsets are called the Base Concepts of EuroWordnet (Rodríguez et al. 1998).

Two synsets from different languages that have the same meaning, are linked together through an InterLingual Index. This ILI record is also attached to one or more Top Ontology concepts, which gives the ILI record some language independent semantic content (Rodríguez et al. 1998).

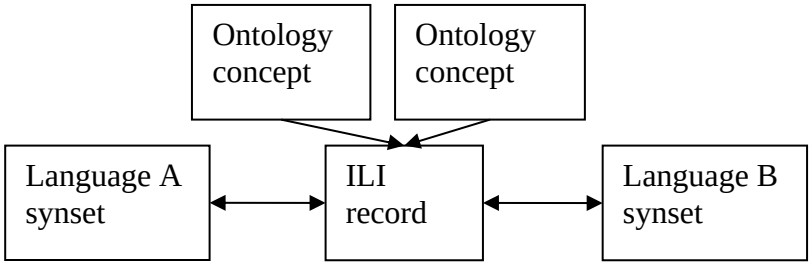


Figure 8. An illustration of how the Interlingual Index works.

The Top Ontology concepts linked to an ILI record must be applicable to everything in all synsets linked to the ILI record.

There are many different versions of the EuroWordnet database. At first, it was built using the software developed for Wordnet (Polaris and Periscope) (De Luca and Lönneker-Rodman 2008), but later on the data have also been transferred to XML (Horák and Smrž 2004) and RFD/OWL (De Luca and Lönneker-Rodman 2008).

By converting the EuroWordnet data to RDF, researchers can take full advantage of the greater compatibility and interoperability this brings. The conversion entailed taking the XML version of

the database and restating it in RDF triples and to establish a database structure using RDF schemas (and the semantics defined by the Web Ontology Language, OWL.) By using tags defined by the RDF schema, computers can make inferences about the data.

Interface and implementation. Unlike Wordnet, Framenet and the Kicktionary, the original EuroWordnet has no online interface and requires researchers to buy licenses to use it. Likewise, the XML and RDF/OWL versions of EuroWordnet are described in articles, but it is difficult to obtain the data and there are no publicly available interfaces.

Challenges and solutions. Two synsets can be linked together in EuroWordnet through the InterLingual Index. The Top Ontology concepts linked to an ILI record must be applicable to everything in all synsets linked to the ILI record. This means that as soon as something is changed in any synset, researchers must ascertain that the Top Ontology Concepts are still applicable to it. This means that modifications to the database are cumbersome. It also means that the person responsible for judging whether the Ontology tagged ILI record really reflects the meanings of all the synsets it is linked to must be linguistically capable of making such calls for all languages.

The original EuroWordnet did not store diachronic information about semantic shifts at all, though there is a later version which integrates the RDF version of EuroWordnet and the Hamburg metaphor database (De Luca and Lönneker-Rodman 2008).

The Hamburg Metaphor Database

Motivation. The Hamburg Metaphor Database stores analyses of metaphors and is intended for computer harvesting. It is integrated with EuroWordnet (Lönneker-Rodman 2008).

Origin and extent of data. The Hamburg Metaphor Database has mainly French, as well as some German metaphors, which are illustrated by 1656 annotated corpus examples. The examples are originally from newspapers, but were gathered and analyzed in a series of master theses at Hamburger University. Each example has been tagged with a source and target domain from the Berkeley Master Metaphor list (Lakoff, Espenson, and Schwartz 1991) while the lexeme (the vehicle for the metaphor) is tagged with data from EuroWordnet. The Berkeley Master Metaphor list is a list of conceptual metaphors (such as TIME IS MONEY[14]) – from this, a list of semantic domain (TIME and MONEY) can be extracted (Lönneker-Rodman 2008).

Database structure. The HMD is constructed as a MySQL database with only three tables: Corpus (data about the original appearance of the example), Thesis (data about the thesis wherein the example had been analyzed) and Metaphor. The Metaphor able has an id, a language and a lexeme. It also has links to EuroWordnet data by identifying both a metaphorical synset and a literal synset in the attributes **synset_met** and **synset_lit**. In the attributes **source_MML** and **target_MML**, the lexeme is connected with a source and target domain from the Berkeley Master Metaphor List (Lönneker-Rodman 2008). The relationships between the tree tables can be seen in Figure 9.

Interface and implementation. The HMD data have been integrated into the OWL/RDF version of EuroWordnet (De Luca and Lönneker-Rodman 2008). It seems to have had an online interface, but this is no longer working[15].

---

[14] Conceptual metaphors and individual semantic concepts are traditionally written in upper case

[15] http://www1.uni-hamburg.de/metaphern//index_en.html

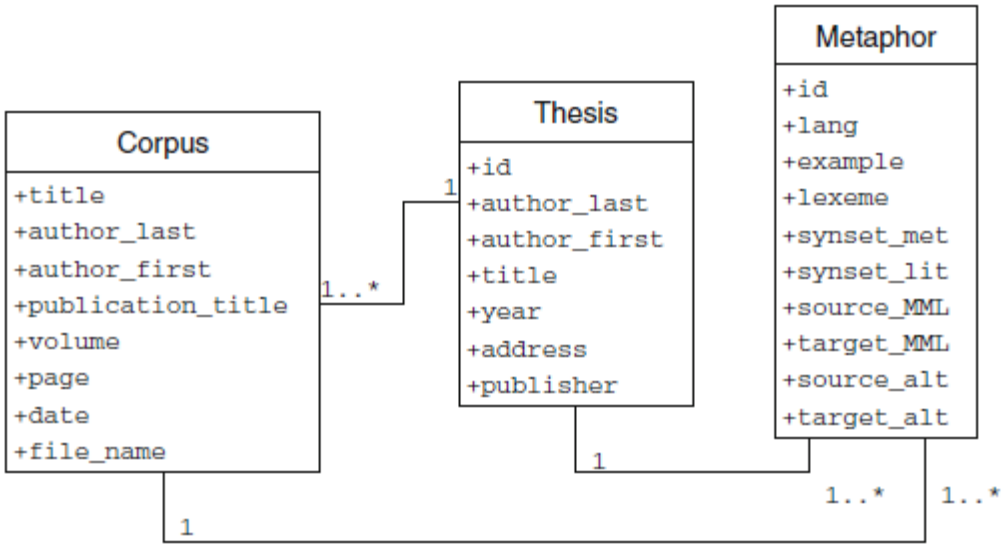


Figure 9. The relationship between the three tables in the Hamburg Metaphor Database, from Lönneker-Rodman (2008).

Challenges and solutions. The HMD is a very simple database and its main value lies in the manual annotation of examples that has been done rather than its data structure. If lexemes – rows in the Metaphor table – had been linked only to abstract source and target domains from the Berkeley Master Metaphor List, the knowledge gained from the database would have been rather shallow. By adding links to EuroWordnet, computers can understand more about how individual lexemes can be used metaphorically.

The HMD has no diachronic information and is unfortunately not publicly available at the moment.

Evaluation of Existing Database Solutions

In the previous sections, I have brought up several databases that are used for semantic research. They all have different foci and implementations.

To do good lexical semantic research on a particular semantic domain, it is desirable to investigate the domain exhaustively, getting both synchronic and diachronic data as well as information on extended meanings and on the link between semantic differences and morphosyntactic differences. While all the existing databases do part of this, none attempts to address all these issues.

One, The Kicktionary, focuses on only one semantic domain. The Hamburg Metaphor Database focuses only on extended meanings. The Catalogue of Semantic Shifts is the only one that has good diachronic data. EuroWordnet and Framenet endeavors to collect data on morphosyntactic patterns.

Some of the older databases, like the first instantiation of Wordnet, used a flat file database model with an index. Framenet started out using SGML markup. Later databases, like the Kicktionary or the Hamburg Metaphor Database use XML or relational databases. There are also versions of Wordnet, EuroWordnet and Framenet using XML databases.

Whatever database model and language is chosen, it must meet several criteria. It must be easy to query; it must be easy to add, delete, manipulate and read the data; it should be as interoperable as possible with existing and future software; and it should make normalization of data easy.

The best way to ensure interoperability is to adhere to the standards worked out by the research community.

The great advantage of using the XML language is the simplicity of the format. If the database interface is coded by hand (rather than being automatically created by the kind of software that exists for SQL), it is easy to access the data. By using XSLT, or other languages, XML can be turned into HTML. Non standardized XML is mostly used as an interim format – like any kind of structured data, there are ways to query it (e.g. by XSLT), but it is not as easy as using SQL queries (for SQL data) or SPARQL queries for RDF data. RDF can be written using a standardized form of XML (there are also other syntaxes) – thus inheriting all the advantages of XML, but also the advantages of RDF. If done correctly, the data will be interoperable with any other RDF resources.

The great advantage of using SQL is the widespread knowledge of how to make SQL queries and the fact that the data will be interoperable with a great deal of software. One example is Django, an open source web application framework which can be used to easily build webpages based on SQL databases. With SQL data, users can also use many different kind of local software, such as Microsoft Access, Open Office Base, Filemaker Pro – all of which have excellent documentation in books and online. This is very important, since small scale research projects must often be maintained by other people than originally built them. If the new caretakers are unfamiliar with the database structure, the database can fall into disrepair and note be used.

SQL and XML (RDF) seem to be the best choices for the Typological Database of Temperature Terms. In the next section, I will first discuss the conceptual model of the Database and then attempt to build two logical database models using SQL and RDF.

Another interesting difference to be noted in the available databases are the different ways of delimiting meaning. I shall call this the taxonomic tree approach and the feature-bag approach. Often, both are used within the same database.

The taxonomic tree approach tries to find the lowest level semantic meme (a sememe) that is used by one of the languages being studied. The feature-bag approach involves finding the lowest sememe that seems useful. The theoretical idea behind two different approaches is shown in Figure 10. In the taxonomic tree approach, two lexemes can be said to have the same semantic meaning, if they are linked to the same sememe (which can be comprised of several sub-sememes), which is illustrated by Lexeme 1 and Lexeme 2 being linked to a sememe involving the sub-sememes sememe X, sememe Y and sememe Z. If X and Y can be found in combination in another concept, this new set will form an independent entry in the dictionary. This is most clearly illustrated by the InterLingual Index used in EuroWordnet. In the feature bag approach, no assumptions are made about the probable groupings of X, Y and Z – instead they are featured as independent concepts in the database. This is illustrated by the way that the Kicktionary uses frame elements to indicate cross-linguistic similarity: the more frame elements are shared between two concepts, the more similar they are. Both approaches have their uses and I will return to them in a later section.

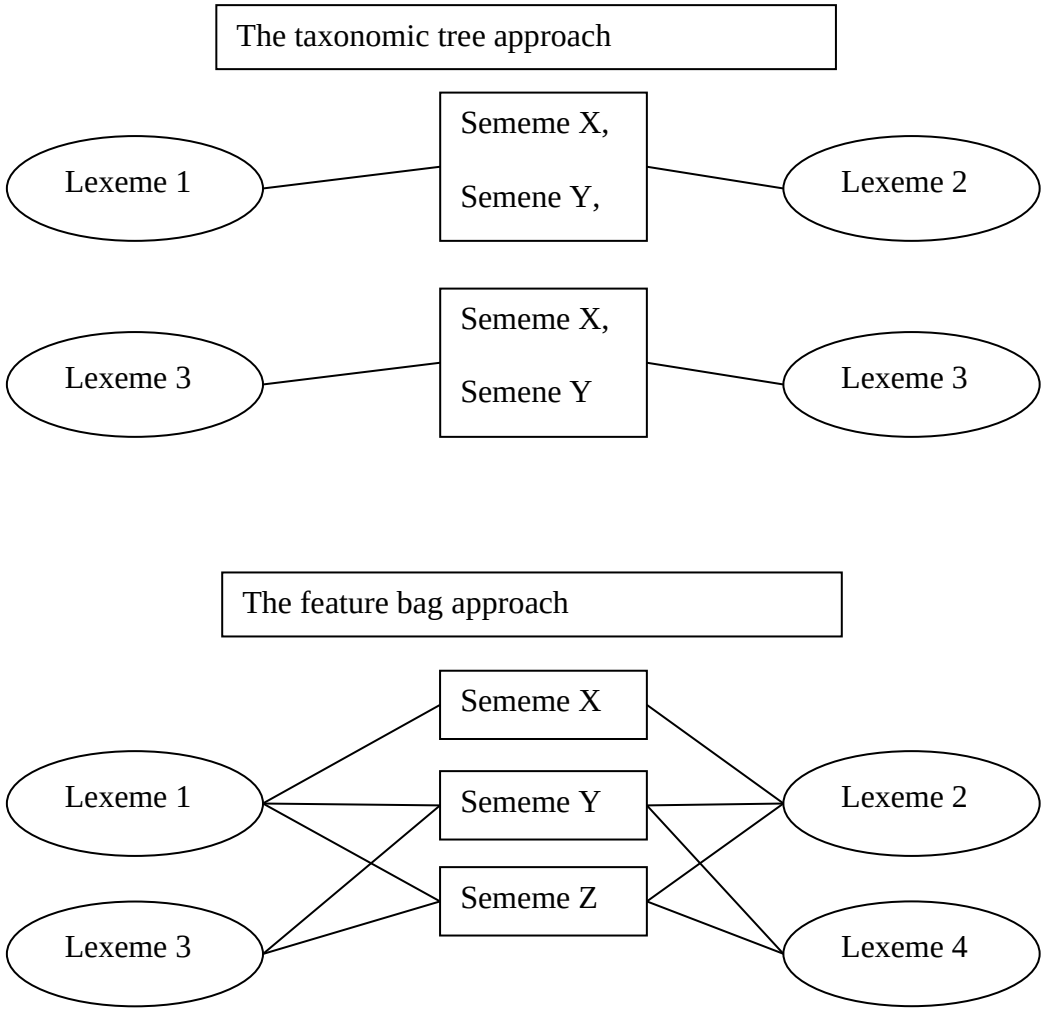


Figure 10. The taxonomic tree approach and the feature bag approach.

Chapter 3: What Kind of Information can Temperature Terms Encode?

A crucial step in database design is typically the establishment of use cases – what the users would like to do with the finished database. In the introduction, I specified a number of general evaluation questions for lexical semantics databases. I revisit these here, restated as statements:

- The database should have a good way of delimiting meaning
- The database should be able to express and comment upon translation equivalences between lexemes in different languages
- It should be easy to implement big changes in the data structure
- It should be easy to use for inexperienced users
- It should be easy to maintain for inexperienced users
- It should be able to store and analyze data on extended meanings
- It should be able to store and analyze data on diachronic semantic shifts
- It should be able to store and analyze data on the interplay between morphosyntax and semantics

Yet these statements are rather vague. In order to make sense of the kinds of data that might need to be stored in the database, I shall now turn to the question: What kind of information are temperature terms encode in the world's languages likely to encode.

The following graph gives an overview of some of the more important semantic concepts, with examples from different languages.

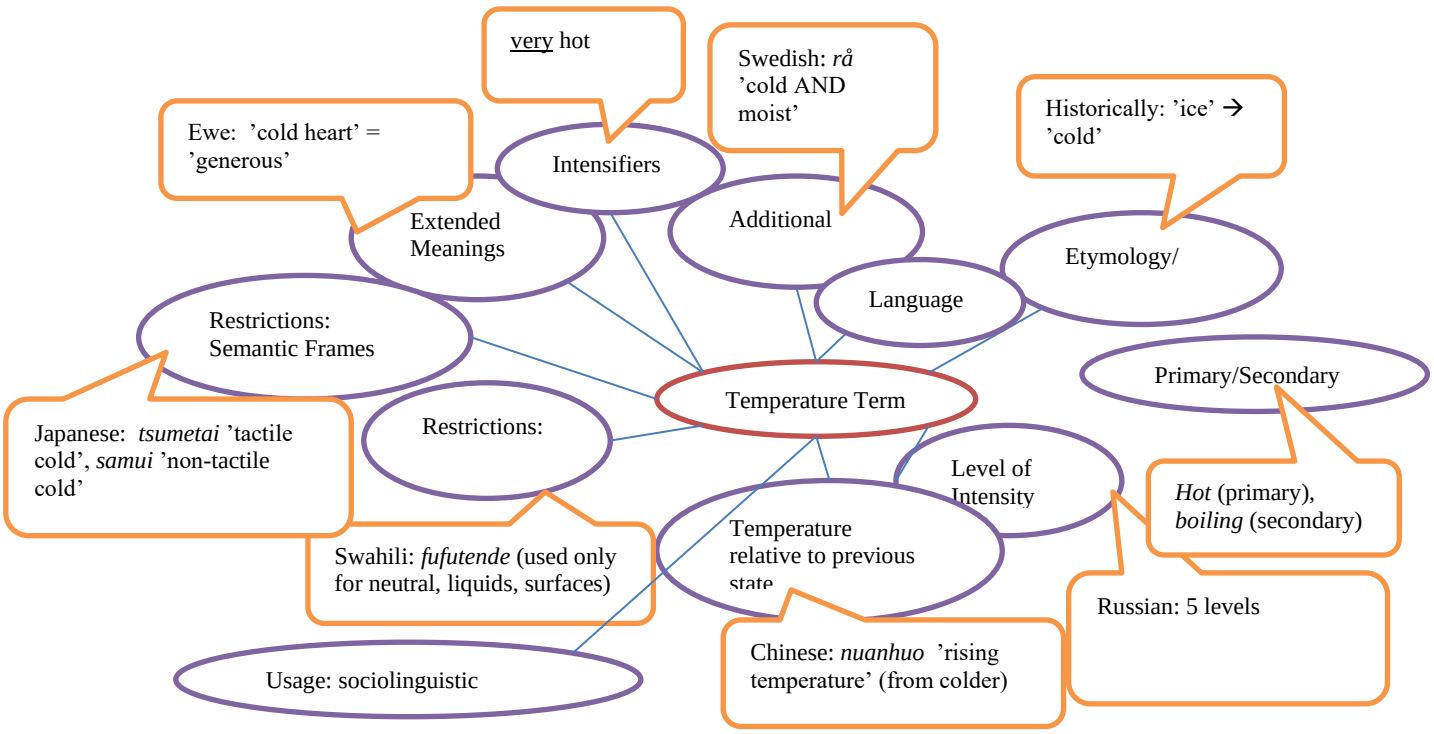


Figure 11. What can temperature terms encode?

Primary or Secondary Temperature Focus

First of all, we can make a distinction between terms that are primarily focused on temperature (such as English *hot)*, and terms that have temperature as a secondary consideration (such as English *boiling* or *sunny*). This is not a clear cut distinction - there are iconic temperature bearing concepts that blur the line, such as *fire*. Further, while languages such as English makes a distinction between primarily heat centered expressions (*hot)* and secondary temperature centered terms (*boiling)*, this is not universal. There is no typological reason to suspect that there might not be languages that talk about the concept of boiling as "water having heat" and the concept of being warm as "a person having heat," for instance.

Basicness

Sutrop (1998) introduced the term basic temperature term. This usage is similar to the basic color term found in the literature on color perception and color naming in different languages (Berlin and Kay 1969). A basic temperature term is of native origin, morphologically simple, psychologically salient and applicable across all semantic domains – it can be used with all kinds of entities and situations.

Level of Intensity

One of the most obvious characteristics of temperature terms is that they vary in intensity and have a scalar quality. Though its precise nature can be discussed, the English term *hot* has a scalar relationship with *cold* and with *warm*. It seems people can intuitively rank temperature terms from coldest to hottest, regardless of the language they speak. English has five temperature terms that can easily be arranged on this scale: *hot, warm, lukewarm, cool* and *cold*, while Mwotlap has two *sɛw* '?hot' and *mɔmjij* '?cold' (Francois forthcoming). Naturally, it is still possible to distinguish between different levels of intensity of temperature in Mwotlap by using intensifiers or similes.

Semantic Frames and Entity Compatibility

Many languages distinguish lexically and/or morphosyntactically between three semantic frames – personal feeling temperatures (*I am warm)*, ambient temperature (*The room is warm*) and tactile temperature (*The water/stone/plate is warm*).

Languages may also have restrictions on which kinds of entities a particular temperature term can be used to describe.

Table 6. Temperature terms and their restrictions in Igbo, Kilba, Yoruba and Ngwo.

	Ngwo				
	Speaker	Food, body parts	Liquids, ice/ snow	Environ-ment, weather, sun	Surfaces, fire
1	efo	dju	yii		dju
2			dju	dju	
3	nom	nom	zebe		nom
4			nom	twon	
5	twon	twon	twon		twon / feh

	Igbo	**Kilba**	**Yoruba**
	Can be used for: Speaker, food, liquids, body parts, environment, weather, wind, surfaces, ice/snow, fire/sun.		
1		shishi'u	tutu
2	oyi		
3			lowooro
4	oku	kwakwaDu	
5			gbona

This is illustrated nicely by Table 6, from Firsching (2010), showing temperature terms in Igbo, Kilba and Yoruba on the one hand and Ngwo on the other. While the three former languages use

the same lexemes for all entities mentioned in the table (speaker, food, liquids etc.), Ngwo has several lexemes than can be used with specific temperatures, such as *vii*, which can only be used with the hottest kind of liquids.

Intensifiers

Other than changing the lexeme, as when *hot* is used instead of *warm* to indicate a very high temperatures, use of intensifiers is very productive in most languages to indicate a change in intensity: *very hot* or *very warm*. In this particular example, *very* is a general intensifier, but English also has a few temperature specific intensifiers, as in *red-hot* or *ice-cold*. Some of these intensifiers might be better described as similes.

Additional Semantic Information

Sometimes, expressions that have temperature as its main semantic focus will still have additional semantic information. When air is labeled *frisk* 'fresh' in Swedish, this means that it is cool and also invigorating. Certain temperature terms, like Ojibwe *dkasin* 'cooling off' indicates a change in temperature, while others describe a static state, as in *cool*.

Etymology and Semantic Shift

The etymology of temperature terms could potentially be very interesting. It would be beneficial to record recurring semantic shifts cross-linguistically, to see which common non-temperature terms (such as ice or fire) are common origins for temperature terms, or if terms for certain intensity levels (e.g. 'cool') typically change diachronically into other intensity levels (e.g. 'cold' or 'lukewarm'). It is also very interesting to know whether a term is borrowed – and if so, to which degree it has been nativized.

Extended Meanings

In addition to diachronic semantic shift, synchronic extended meaning is also something that should be recorded. Are there cross-linguistic similarities in the target domains (KINDNESS*?* ATTRACTION*?*) that metaphorically or metonymically used temperature terms can be used for. Is it, for instance, common cross-linguistically that *warm person* means ‘a kind person’ or *a hot guy* ‘an attractive guy’.

Usage

The frequency of use for temperature terms is an important piece of information. For many languages, it is not possible to get a hold of corpus data to verify such information, however. Another, related piece of information is the saliency of the term – how easy is it for speakers to recall it when asked to list temperature terms. There should be a way in the database to record this.

Language

Other than the language name and ISO code of the language that a specific term is from, it is good to know where it is spoken, the number of speakers it has and which language family it belongs to.

Morphosyntactic Behavior

The morphosyntactic behavior of temperature terms might or might not be the same as that of terms describing similar information (thickness, height, sharpness). It is also possible that different word classes are used for different kinds of temperature terms.

Examples

The different kinds of ways that temperature terms can be used should be illustrated by examples. When it comes to examples of extended meanings, both a literal and the actual translation should be included and it should always be possible to enter morpheme by morpheme breakdowns of the examples.

Sources

Ideally, it should be possible to reference every single piece of information entered about a particular temperature term, just as it is in an academic paper. Certain kinds of sources, such as examples attributed to speakers that wish to be anonymous, might need to be opaque.

Chapter 4: The Conceptual-Logical Design

In the previous sections, I have discussed several different databases used in lexical semantics research, as well as the requirements for a lexical semantic database of temperature terms. In chapter 5 on page 80on page, I concluded that XML (RDF) and SQL were the best candidate languages to use for the database. In the following section, I intend to sketch first a conceptual model using RDF and later a conceptual model using SQL. The models will sometimes include some things (such as data types) that typically belong to the logical design of a database – but since, in practice, the logical design varies depending on the DBMS, I will not discuss it in depth.

An RDF Conceptual Model

This section discusses the structure of the RDF conceptual database model which is illustrated in Appendix 1.

Whenever possible, existing RDF standardized tags should be used. The most frequent examples in the sections to follow will be when the Dublin Core (dc) is used - The Dublin Core is a set of predefined properties for describing documents.

The Lexeme. Conceptually, the central part of the RDF database would be the Lexeme. It has a written form and an English translation. A lexeme must also belong to a language, but while the written form and the translation can take the form of string literals, a language is a resource, which should get its own URI.

A Language resource should have a name, a string literal. It also belongs to a Language family, which can be viewed as a resource. Both the language name and the language family name could have URIs directly connected to the Multitree or Ethnologue online language resources, which

have language specific URIs. Another solution might be to let the URI be based on the local namespace for the project and then link this to several language resources and language family resources, such as Ethnologue and Multitree.

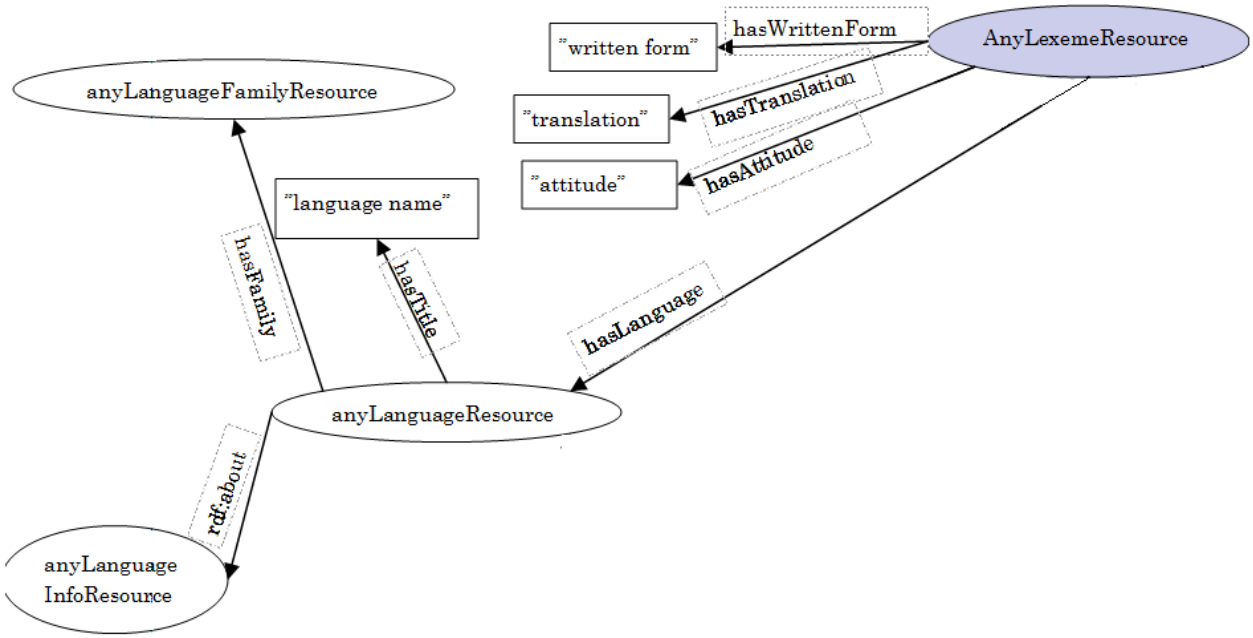


Figure 12. An excerpt from the RDF conceptual model.

Sources, comments and descriptions. Any resources or RDF triple in the RDF database should be able to be connected to a source and to a comment field and a description field. The former should be a resource in its own right, with information on title, publisher, journal title, page range, and authors. Authors are resources and should have their own URIs. It is important which order the authors of a book or article come in – which necessitates the use of an RDF container. By using a container, it is possible to give place numbers to the resources that make up the container – an RDF:Sequence. The first author of a published work will get the number one, the second will get the number two and so on.

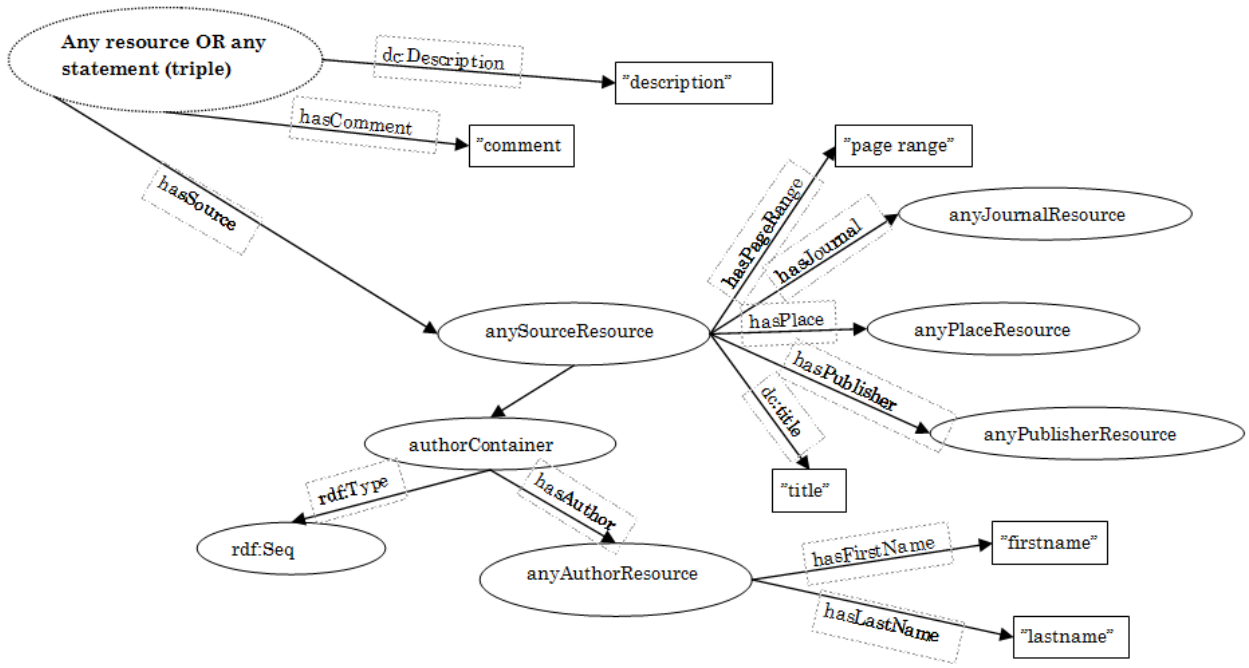


Figure 13. An excerpt from the RDF conceptual model.

Primary or secondary temperature focus. Until data are added to the database, it is difficult to know to what extent the issue of primary or secondary temperature focus will show up. It is an important distinction to remember for future redesigns of the database, but I decided to let any information on primary or secondary temperature focus be gathered in text form in the comment texts about the lexemes.

Basicness. Whether a language is basic or not might have been established by an individual researcher, using the rather vague definitions available in the literature (Sutrop 1998) – but it is equally possible that its basicness status is as yet undecided, or that no such data **are available.** Since it should only be possible to choose from a limited set of basicness statuses (yes, no, maybe or unknown), this property cannot be a string literal, but must be represented by a set of clear resources with their own URIs.

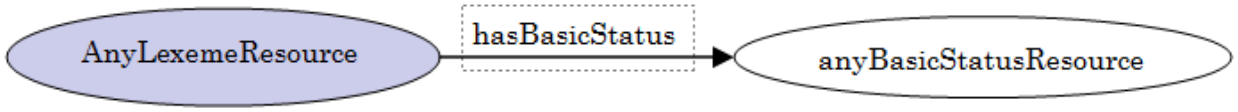


Figure 14. Basicness in the RDF conceptual model.

Level of intensity. Within the same language, there might be several parallel scales of intensity. It is conceivable that a language might have, for instance, special lexemes denoting hot and cold water, and that it will only make a two level distinction in intensity for water, while the same language has a five level intensity distinction for temperature words for room temperature. Therefore it would not work to have a predicate hasTotalIntensityLevel for each language (subject) and then a literal object denoting the intensity level of the particular lexeme. Instead, I decided to represent this by having the lexeme be the subject, creating a hasIntensityLevel predicate and a string literal in the form *x/y* where *x* represented the intensity level of the term in question, while *y* represented the total number intensity levels. The *y* would be different in different languages. The lower the number, the colder the term. This would make English *hot* a 5/5, since it represents the warmest possible level (that is lexicalized without intensifiers (very hot) or similes (boiling hot) out of the five levels that are lexically represented in English. Likewise, *cold* would be 1/5, *cool* would be 2/5, *lukewarm* would be 3/5 and *warm* would be 4/5.

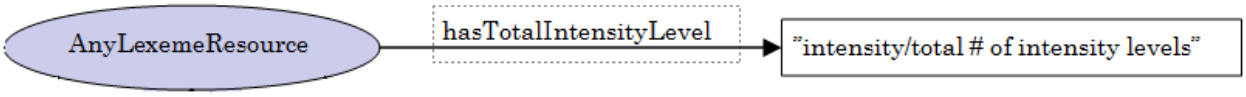


Figure 15. Level of intensity in the RDF conceptual model.

Semantic domains and entity compatibility. Many languages have morphosyntactic or lexical differences in the temperature terms used for tactile temperature experiences (temperature when you touch something), non-tactile temperature experiences (like room temperature, felt with the

entire body) and experiencer temperatures (when a person feels warm or cold). In order to capture this, I envision a set of SemanticDomainResources specifying what kind of temperature experiences a particular lexeme can encode. This should be an open set, which users can add new semantic domains to.

The set is envisioned as a hierarchical taxonomy, such that if a lexeme can be used for things higher up in the taxonomy, it can also be used for things lower in the taxonomy. Any lexeme resource that can denote tactile temperature experiences, can also denote water temperature or metal temperature, since water temperature is a kind of tactile temperature. But if the database states that a lexeme can be used with water temperatures – this does not mean that it can be used with all other kinds of tactile temperatures (like the temperature of surfaces, metal objects, food etc.). Each SemanticDomain resource can be the subject in a triple with the predicate hasMorphoSyntaxNote and a string literal object, where users can describe its morphosyntactic pattern and if there is anything of special note in the morphosyntax of temperature terms in the language.

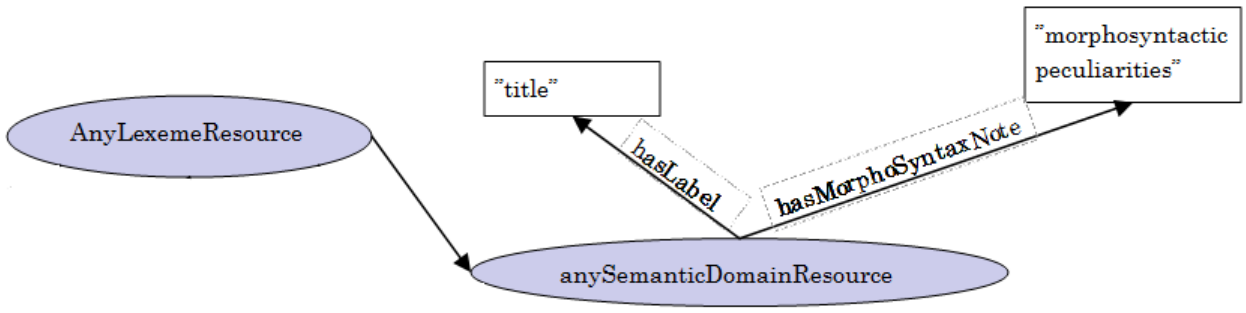


Figure 16. Semantic domains in the RDF conceptual model.

Intensifiers. In the first conceptual RDF design, I decided against adding data structures to encode the way intensifiers are used with lexemes. It would have been easy to add functionality to encode a list of intensifiers – but I was uncertain how useful this would be. I felt that although

the question of intensifiers was very interesting, I did not know enough about their potential cross linguistic use with temperature terms to ask users the right questions.

Additional semantic information. In order to capture that the English expression *sultry* not only means hot, but also humid, and that the Swedish term *rått* not only means cold and humid but also invariably an unpleasant kind of cold, *I* created two RDF predicates: hasAdditionalSemanticFeature and has Attitude. Both take string literals as objects.

Etymology and semantic shifts. While I would have liked to incorporate a structure similar to the Catalogue of Semantic Shifts in TDTT, I am uncertain it presents the best way to address semantic shifts in a very limited semantic domain, such as temperature. Since not much has been done in the diachronic study of temperature terms, I decided that a more qualitative than quantitative strategy in information gathering might be ideal. With this in mind, I created a structure that was very similar to the RDF structure for Semantic Domains: Lexeme resources could take the predicate hasEtymology and an Etymology resource as object. An Etymology is a particular meaning that the lexeme once had, other than temperature meaning. This would be an open set of resources – new Etymology resources could be created at any time by users.

Etymology resources can also come from other languages – therefore I envisioned a link between them and the language resources.

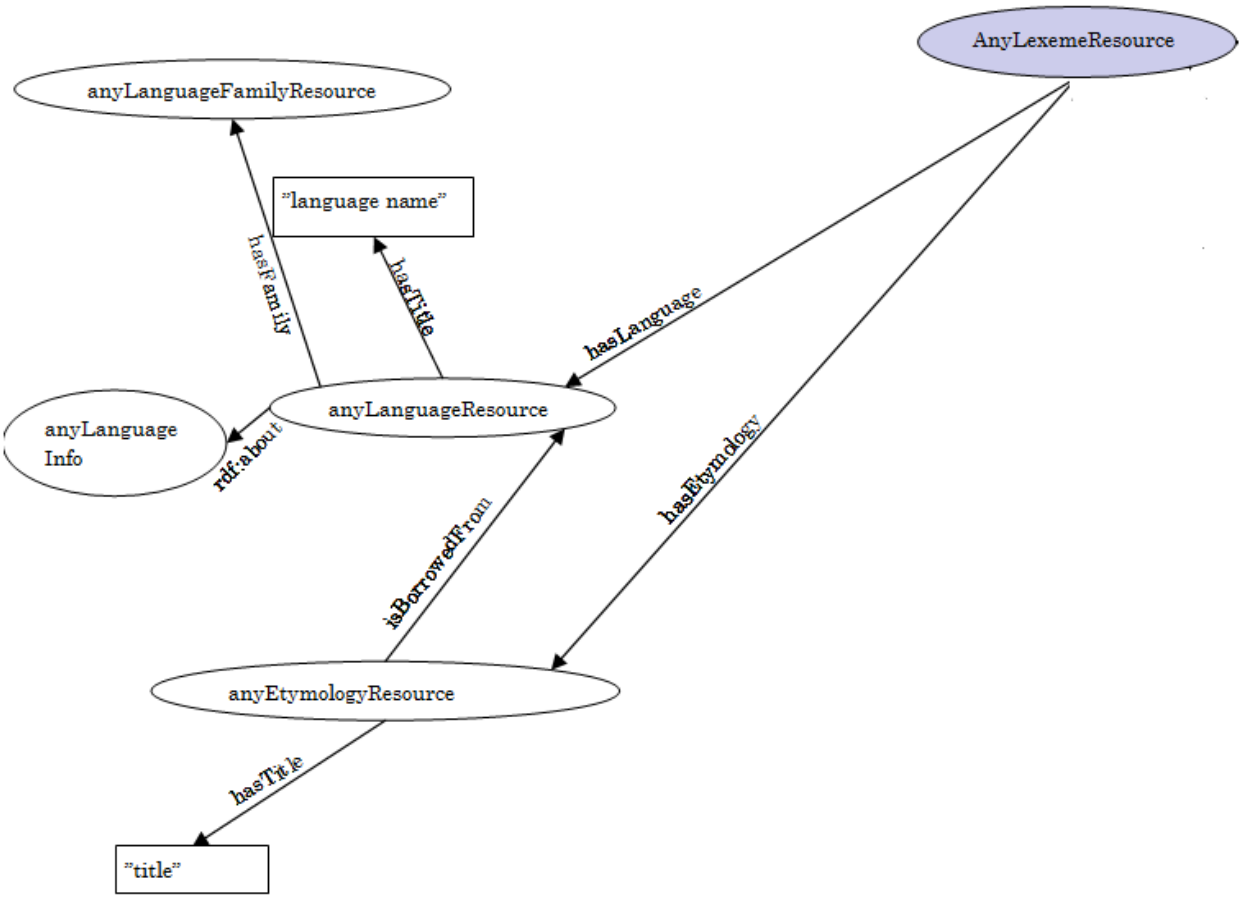


Figure 17. Etymology in the RDF conceptual model.

Usage. Since few published studies of temperature terms have usage statistics or reports from saliency experiments, I decided that this data could be stored in the comment to the lexeme (subject: lexeme resource, predicate: hasComment, object: string literal).

Extended meanings. The Hamburg metaphor database analyzed metaphors by identifying their source and target domains in Lakoff, Espenson and Schwartz (1991). In TDTT all metaphors will have the same source domains and I would like to try to analyze their target domains with a higher level of level than the Berkeley Master Metaphor List makes possible. Unfortunately, until I have data to analyze, I don't know which target domains are likely to appear – a typical

problem in the design of typological databases. Therefore, I decided to adopt an approach that is very similar to the one I use for the etymological data and semantic domain data.

Each lexeme resource can be associated with many TargetDomain resources. At first, there will be no rules about how to express these target domains – users can input anything. A future reanalysis of the data will most likely change this aspect of the database and try to collapse some of the user created target domains.

Examples. The database should be able to store examples of both how a lexeme is used in a particular semantic domain and how it is used in metaphors or metonyms. To do this, the RDF database employs reification. Instead of taking a simple resource as the subject of a triple, it takes an entire triple (an entire RDF statement) as the subject. The statement that a particular lexeme has a particular semantic domain (or a target domain) becomes the subject, the predicate is hasExample and the object is an Example resource.

I envision each string literal of each example sentence as being attached as a string literal to an Example resource. This way, the same Example resource can be used to illustrate many kinds of semantic domains and/or target domains.

The Example resources can also have string literal translations and be linked to SyntacticFunctionResource. The latter would be a closed set of resources denoting different syntactic functions, such as predicate, attributive – it can also be left blank, since these are not necessarily distinctions that are meaningful in all languages.

It should also be possible to produce IGT of each example. Since it is important which order the words in the example sentence comes in, the Example resource would have to be linked to a word container, which is a sequence. The WordContainer could then be linked to any number of

Word resources – and they would be associated with a number depending on where the word is in the sentence. To produce the actual IGT, Word resources can have several string literals linked through the predicates hasLexemeLevel, hasMorphemeLevel, hasInterpretation and hasTranslation.

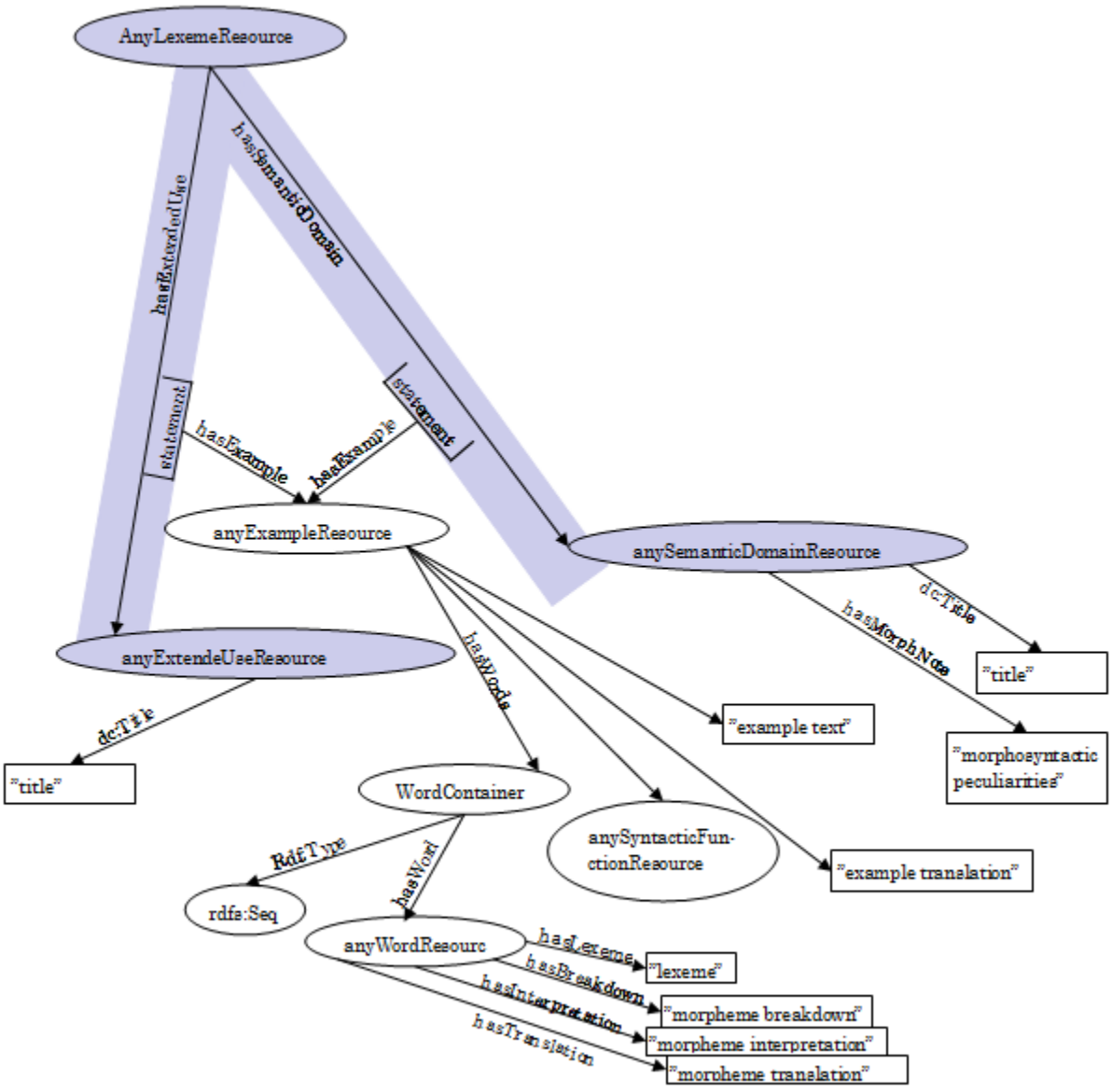


Figure 18. Extended uses, Semantic Domains and examples in the RDF conceptual model.

An SQL Conceptual Database Model

The following is an SQL conceptual database model for TDTT. The whole model can be seen in Appendix 2.

Some of the illustrative tables below have an undulating bottom line – this is to show that not all the attributes of the table are shown. One too many relationships are represented by an arrow (where the tail of the arrow connects to the "one" table and the arrowhead to the "many" table.

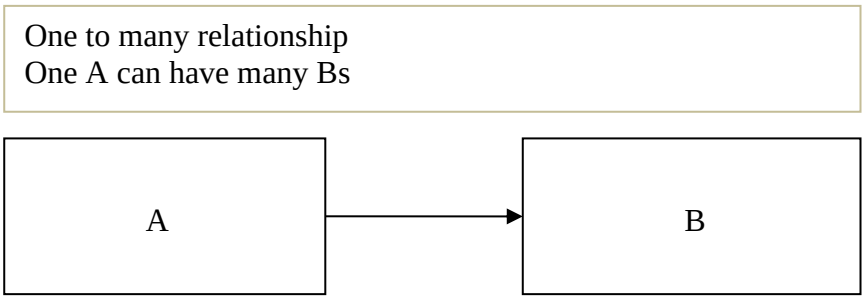


Figure 19. One-to-many relationships.

The lexeme. I decided to make the temperature terms themselves – the lexemes – the center of the database design and started by postulating a table called **Lexeme**, which would have a number of attributes and a number of relations to other table. Naturally, the first attribute associated with the table **Lexeme** is its automatically generated ID (and primary key), **lexeme_ID**. As we discuss the other information below, we will see how the table was further populated with more attributes, as well as relations with other tables. The lexeme itself could be entered in a text field in the attribute **lexeme** and a translation in a text field **free_translation**.

Table 7. The (partial) Lexemes and Languages tables.

Lexemes		
Name	**Type**	**Primary Key, Foreign Keys**
lexeme_ID	Int	PK
lang_code	Varchar(7)	FK (Languages)
lexeme	Text	
free_translation	Text	

Languages		
Name	**Type**	**Primary Key, Foreign Keys**
lang_code	Varchar(7)	PK
language_name	Text	
language_family	Text	
general_comments	Text	

Information about which language the lexeme came from could have been a simple attribute in the **Lexeme** table, but since language information might come into play at other stages as well, a second table – **Languages** – was created and the primary key of the **Languages** table, **lang_code**, was inserted as a foreign key in the **Lexemes** table. In addition the Languages table had a **language_name** attribute, a **language_family** attribute and a **general_comments** attribute.

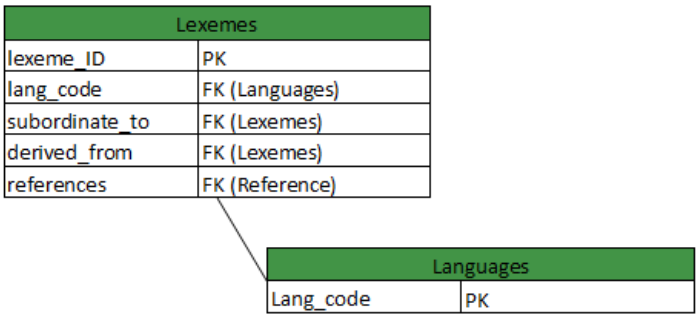


Figure 20. The Lexemes and Languages tables, with primary and foreign keys.

Sources. In a relational database, one way to tag every single attribute with source information would be to create a **Reference** table and then create an additional sister attribute with the source information and add it after every data attribute. This is certainly cumbersome and tiring, since a lot of the data probably comes from the same source.

An alternative is to create a new linking table, let's hypothetically call it the **Source_Instance** table, which links every single data attribute in the other tables together with a row in the Sources table. This hypothetical table would need to use a combined primary key of the table the data came from, together with the all the information in the data attribute itself.

Some DBMSs have a problem implementing combined primary keys, however. Also, this creates a strikingly complex data infrastructure.

Interestingly, this problem would be far easier to solve in an XML (RDF) database.

As it is, I used a much cruder solution. Every table, that is not a linking table (i.e which primary focus is linking two other tables together) , has a **source** attribute, which links it to one of the references in a **Reference** table. This means that only one source can be given for each row in each table and that one cannot give different sources for different bits of information in the same table. This is undoubtedly problematic.

The **Reference** table has an attribute **authors**, which, while being a text field, is still the primary key of the table. It should contain the authors' name and the year of publication, in the following manner: *Smith, Jane. 2010.* Following academic tradition, if the same author having several publications each year, they are suffixed with letters, such as *Smith, Jane. 2010a* and *Smith, Jane. 2010b.*

Another attribute in the **Reference** table is **title** which should contain the title of the publication. The attribute **publisher_place** should contain, if the reference is a book, the name of the publisher and the page it was printed at. In addition there is an attribute called **journal_page**, which should contain the name of the journal, if applicable, as well as the starting page number, if the work is an article. These three attributes were all text fields.

Table 8. The Reference table.

Reference		
Name	**Type**	**Primary Key, Foreign Keys**
authors	Text(50)	PK
title	Text	
publisher_place	Text	
journal_page	Text	

Primary or secondary temperature focus. In the first analysis, I decided that any temperature related term could be included, with no distinctions made. The reason for this was simple – I did not have enough data to formulate rules on which was to be included or not. It was assumed that a later reanalysis would have to remove some entries for being irrelevant (maybe remove *boiling* and *steaming* but keep *hot)*, and add others. Therefore, no special attribute was created for primary or secondary temperature focus.

Basicness. Basicness is, ideally a binary state – a term is either basic or it isn't. However, the research community has yet to establish firm rules for what exactly would constitute a basic temperature term – and data might need to be added to the database while research into the matter is still ongoing. Sometimes, the researcher might not feel comfortable enough to make the judgment whether the term is basic. This prompted a design where the question of basicness could be answered by "yes," "no," "maybe" or a blank answer, indicating that the information

was not available. Since each lexeme has one and only one basicness status (a 1:1 relationship), **basic** was designed as an attribute to the lexeme table. The attribute was a list with the four possible entries: “yes,” “no,” “maybe” or blank.

A comment field was also created as an additional attribute in the lexemes table, to allow researchers to enter as much text as they wished to discuss the issue: **basic_comment**.

Table 9. A partial Lexemes table.

Lexemes		
Name	**Type**	**Primary Key, Foreign Keys**
basic	Tiny Text	
basic_comment	Tiny Text	

Level of intensity. While one language might have a total of five levels of intensity, another might only have two. It was decided that this would be represented in the form *x/y* where *x* represented the intensity level of the term in question, while *y* represented the total number intensity levels in the language. The lower the number, the colder the term. This would make English *hot* a 5/5, since it represents the warmest possible level (that is lexicalized without intensifiers or similes) out of the five levels that are lexically represented in English. Likewise, *cold* would be 1/5, *cool* would be 2/5, *lukewarm* would be 3/5 and *warm* would be 4/5. Since intensity is a characteristic of each temperature term, it was made an attribute of the Lexeme table – the **intensity_total** attribute.

Naturally the attribute described above was accompanied by a sister *comment on basicness* attribute, which was a text field of unlimited size where researchers could enter data.

Given the figure 4/6 as the intensity level for a particular lexeme is less than satisfying, however – for one thing, it is unclear whether the term is in the warm or cold zone! To solve this, an additional attribute to the Lexeme table was created – **temperature_zone**, a choice between *warm zone, neutral zone* and *cold zone*.

In addition, an attribute **intensity_in_zone** was created, which worked similarly to the **intensity_total** attribute, but was only focused on one temperature zone at a time. As an example, English has two warmer terms – *hot* and *warm* – and *hot* was thus labeled with **intensity_in_zone** 2/2 and *warm* was labeled 1/2, where "2" represents the number of terms in the warm zone.

Temperature_zone and **intensity_in_zone** were made attributes of the Lexeme table.

There are instances where one temperature term might in intensity and coverage of semantic frames, subsume another. An example of this is the Swedish term *het* 'hot', which is subsumed by *varm* 'warm' – i.e, anything that is *het*, can also be described as *varm*: *varm* covers several more degrees of intensity than *het* does (Koptjevskaja-Tamm and Rakhilina 2006). In order to capture this, I created an attribute to the Lexeme table: **subordinate_to**. This attribute takes another lexeme's primary key as its value – which means that it refers back to its own table.

Table 10. A partial Lexemes table.

Lexemes		
Name	**Type**	**Primary Key, Foreign Keys**
intensity_total	Tiny Text	
intensity_in_zone	Tiny Text	
temperature_zone	Tiny Text	
temperature_zone_comment	Text	

Semantic domains and entity compatibility. Based on the work done by Koptjevskaja-Tamma and Rakhilina (2006), I decided to construct a taxonomy of semantic domains, based on the top node **temperature**, with the three daughter nodes **tactile**, **non-tactile** and **experiencer**. Every lexeme could be tagged with one or more of these nodes – though tagging a lexeme with both the top node **temperature** and a daughter node would be redundant, since the top node subsumes all the other nodes. There are many lexemes that have far more fine grained distinctions than tactile, non-tactile and experiencer temperatures and in order to let users add as many new levels to the taxonomy as they would wish, a new table **Semantic_Domains** was created, which was linked to the table **Lexeme** through the link table **Semantic_Domain_Instance**. The **Semantic_Domain_Instance table** had a many-to-one relationship with the **Lexemes** table and the **Semantic_Domain** table.

The information associated with a **Semantic_Domain** is an automatically generated ID – attribute name: **semantic_Domain_ID,** which of course also is the primary key of the table – the label we give it – attribute name **label** – a comment or description – **comment**.

The information associated with the semantic domain of a particular lexeme – i.e a semantic domain instance – is a primary key (**sdiID)**, as well as the IDs of the particular lexeme and the particular semantic domain – **lexemeID** and **semantic_DomainID**. These two are foreign keys. Since the morphosyntactic patterns for temperature terms might be interesting topics for research, two text field attributes – **morphosyntacticPattern** and **commentsOnNoteworthiness** – were added. The former was to capture the morphosyntactic behavior, in both attributive and predicative use, that the lexeme had when it was used in the semantic domain in question.

Table 11. The Semantic_Domain_Instance Table.

Semantic_Domain_Instance		
Name	**Type**	**Primary Key, Foreign Keys**
sdiID	Int	PK
semantic_domain_ID	Int	FK (Semantic_Domain_ Instance)
lexeme_ID	Int	
comment	Text	
morphosyntactic_pattern	Text	
comments_on_noteworthiness	Text	

(To give an example from English – it is possible to say *I am warm* and *The room is warm* and *A warm room*, but not *A warm I*). The latter attribute is a comment field specifically dedicated to comments about the morphosyntactic behavior of the lexeme in this context – was the pattern noteworthy or different from the way other similar concepts were morphosyntactically expressed?

A general **comment** attribute, as a text field, was also created for the **Semantic_Domain_Instance** table.

Table 12. Semantic_Domains table.

Semantic_Domains		
Name	**Type**	**Primary Key, Foreign Keys**
semantic_domain_ID	Int	PK
label	Tiny Text	
comment	Text	
standard_example_sentence	Tiny Text	

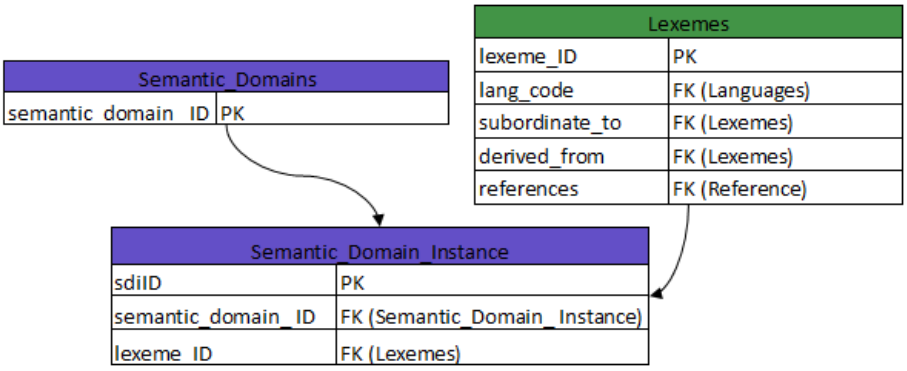


Figure 21. Semantic Domains, Sematic_Domain_Instance and Lexemes tables.

Intensifiers. In the first conceptual design, information on intensifiers did not get an attribute of its own. I did not have enough information about intensifiers and temperature terms to know which questions were interesting to ask.

Additional semantic information In both example 1 from English and 2 below from Ewe, there is additional semantic information other than plain temperature information given. In 1), *sultry* means both 'hot' and 'humid' and in 2) *dzo* means 'fire', but is also the way heat is expressed: cf. 3) (from Ewe) where *dzo* is used again, but this time to mean actual fire.

1) A sultry day
   A hot and humid day
2) A*f*é-á *me* *dze* *dzo*
   House-DEF *containing.region* *contactfire*
   The house is hot (inside)
3) Afé-á *dze* *dzo*
   House-DEF *contactfire*
   The house is on fire

But there is a difference between 1) and 2) but the exact nature of this difference is hard to pin down without data from more languages. In the first conceptual design, additional semantic information was made a plain text field attribute of Lexemes, called

additional_semantic_feature, with the idea to gather text information on additional semantic information for reanalysis and redesign of the database at a later stage.

A related, but slightly different, kind of information, is whether a particular lexeme is always a positive or negative evaluation. The Swedish term *råtт* means 'cold and humid', but could never be used to characterize a pleasant cold and humid experience: it is inherently negative. The attribute **attitude** (free text field) was created as a place to provide this kind of information about a lexeme.

Table 13. A partial Lexemes table.

Lexemes		
Name	**Type**	**Primary Key, Foreign Keys**
attitude	Text	
additional_semantic_feature	Tiny Text	

Etymology and semantic shifts. The Zalizniak (2008) article showcases the database of semantic shifts – a database which solely concerns itself with diachronic and synchronic shifts of meaning. The database has very intricate and detailed information on semantic shifts. While the ideal solution would be to attach such a module to the Typological Database of Temperature Terms as well, it is not viable give the scope and time available and a simpler model had to be settled for.

While each lexeme has a unique etymological history, it is primarily interesting, from a typological perspective, to discover any recurring patterns to the semantic change of temperature terms. Therefore, I created a short list of etymological semantic origins, containing the words *ice*, *fire*, and *snow*. This list is infinitely expandable by users, as new etymological origins appear when new terms are studied. It is also of great importance which other language, if any, the word

was borrowed from and there are probably additional data on when and how this happened that should be encoded in the database. I created an additional table, **Etymology**, which was linked to the **Lexeme** table through the linking table **Etymology_Instance**.

The **Etymology** table has an ID and primary key, **etymologyID**, a **label** attribute and a **comment** attribute where the information can be expanded.

The **Etymology_Instance** table contains an ID and primary key, **ei_ID** and foreign keys to the related tables mentioned above – **lexemeID** and **etymology_Instance_ID**. If the lexeme was borrowed from another language at some point in time, this information was included by linking in the language from the **Languages** table, through a foreign key **langCode** (masked with the attribute name **borrowed_from** in the **Etymology_Instance** table). In addition a text field **comment** attribute was added, as well as a **source** attribute, which consisted of a foreign key linking to the table **References**.

But the **Etymology** and **Etymology_Instance** tables are intended to primarily deal with diachronic information. It is very possible that there will be lexical items, like the English expression *red hot* which are derived from other contemporary expressions, in this case *hot*. In order to link such lexemes together, I created an attribute **derived_from** in the Lexemes table, which had the primary key of another lexeme as its value. This kind of loop self-reference in a table should in theory be handled by any DBMS that implements mySQL, but in reality Microsoft Access, for instance, has problems with this.

Sometimes, a lexeme might be a clear derivation from an existing non-temperature term and a researcher might want to give the origin of the term, without creating a specific entry for it. This is implemented through an additional attribute in the Lexemes table: **derived_from_unique**.

I also created a free text field attribute, **derived_comment**, where users could expand upon this.

Table 14. A partial Lexemes table.

Lexemes		
Name	**Type**	**Primary Key, Foreign Keys**
derived_from	Int	FK (Lexemes)
derived_comment	Tiny Text	
derived_from_unique	Tiny Text	

Table 15. The Etymology table and Etymology Instance table.

Etymology			
Name	**Type**	**Size**	**Primary Key, Foreign Keys**
etymology_ID	Int	2147483647	PK
label	Tiny Text	255	
comment	Text	65,535	

Etymology_Instance			
Name	**Type**	**Size**	**Primary Key, Foreign Keys**
ei_ID	Int	2147483647	PK
lexeme_ID	Int	2147483647	FK (Lexemes)
etymology_ID	Int	2147483647	FK (Etymology)
borrowed_from	Varchar(7)	7	FK (Languages)
comment	Text	65,535	
source	Tiny Text(50)	50	FK (Reference)

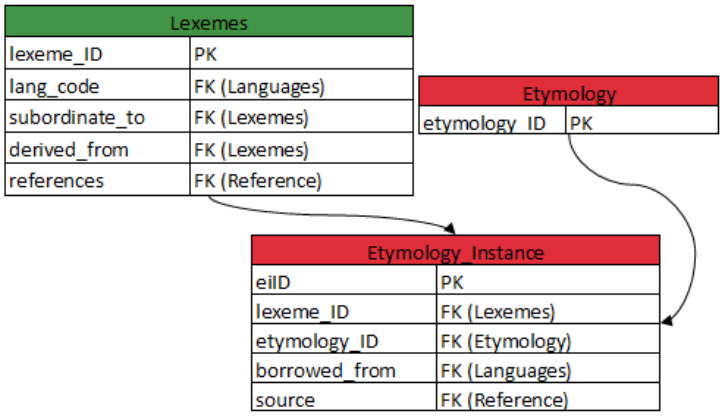


Figure 22. The Lexemes, Etymology_Instance and Etymology tables.

Extended meanings. One of the biggest challenges in this database project was deciding how to encode extended semantic meaning – primarily metaphors and metonymies, but also idioms – in a good queryable form without losing semantic content. Semanticists interested in cross-linguistic use of metaphors would wish to know if several lexemes in several languages were used for the same kind of extended meaning – yet something as complex as a metaphor is rarely exactly the same in different languages. *A warm smile* in English and *Ett varmt leende* 'a warm smile' in Swedish are very similar and both denote a smile expressing affection and friendliness, but a detailed analysis might find differences. *Warm kisses* and *Varma kyssar* are not really the same, since there are two words in Swedish that can be translated as *kiss – kyss* which is typically a kiss lovers share and *puss* which is more akin to *a peck on the cheek/mouth* and which is typically a kiss parents and children can share. Thus, the database infrastructure must be able to capture both the similarities and differences of the two expressions *warm kisses* and *varma kyssar*.

The first conceptual design was very consciously not equipped to deal with this level of detail – in order to know what needed to be captured by the system, a fair amount of data had to be analyzed. Therefore, a preliminary rough hewn database structure was conceptualized: one in which each metaphorical usage of a term had to be linked to a target domain – either a target domain that already had been established, or a new one that the user created. In this way, a list of possible target domains would be generated, with the hope that this data could be reanalyzed and prompt a conceptual redesign of the metaphor module of the database.

To implement this, two new tables were designed: **Target_Domain** and **Target_Domain_Instance**.

The **Target_Domain** table had an ID and primary key, **target_Domain_ID**, a text field attribute **label** and a text field attribute **comment**. The **Target_Domain_Instance** table had an auto generated ID and primary key, **tdi_ID**, and had two foreign keys **target_Domain_ID** linking to the table **Target_Domain** and **lexemeID** linking to the table **Lexeme**. It also had a **comment** attribute, a text field.

Table 16. The Target_Domains and Target_Domain_Instance tables.

Target_Domain_Instance			
Name	Type	Size	Primary Key, Foreign Keys
tdi_ID	Int	2147483647	PK
target_domain_ID	Int	2147483647	FK (Target_Domains)
lexeme_ID	Int	2147483647	FK (Lexemes)
comment	Text	65,535	

Target_Domains			
Name	Type	Size	Primary Key, Foreign Keys
target_domain_ID	Int	2147483647	PK
label	Text	255	
comment	Text	65,535	

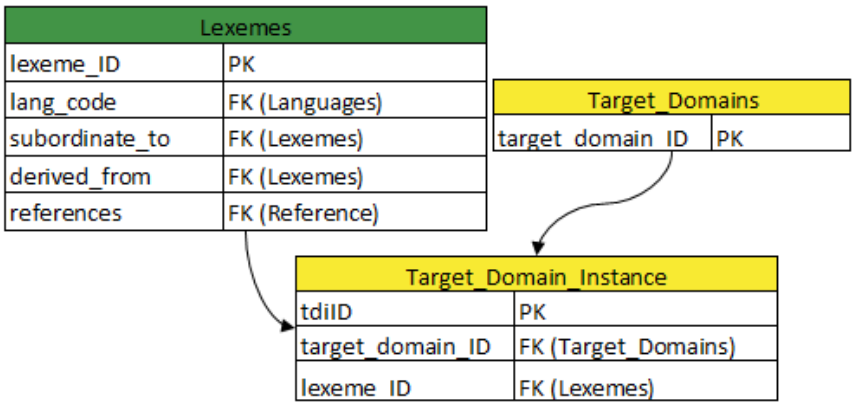


Figure 23. The Lexemes, Target_Domains and Target_Domain_Instance tables.

Usage. Since few published studies of temperature terms have usage statistics or reports from saliency experiments, it was decided that this data would not be given an attribute of its own, but could be placed in the attribute **general_comment** in the Lexeme table.

Morphosyntactic behavior. It would not be enough to have attributes of the Lexeme table which purpose was to document the morphosyntactic behavior of the lexeme.

Instead, the differences in the morphosyntactic behavior of the lexeme must be documented for each specific semantic domain – and this was accomplished by adding the attribute **morphosyntacticPattern** and the related attribute **commentsOnNoteworthiness** to the **Semantic_Domain_Instance** table. Two additional attributes were added to the Lexeme table as well: **wordclass** and **wordclass_Comment**.. The former had an enum list of values (*adjective, verb, noun, participle, adverb, other*) and the latter was a free text field.

Examples. Examples of lexemes should be given in the context of which semantic domain or which metaphorical usage they can be used to express – to ensure this, I created new tables, namely **Metaphor_Example** and **Domain_Example**, which were linked to each **Semantic_Domain_Instance** and **Target_Domain_Instance**. **Semantic_Domain_Example** and **Target_Domain_Example** have exactly the same structure, as can be seen below:

Table 17 The Metaphor_Example table.

Domain_Example	
domain_example_ID	PK
instance_using_example	FK (Semantic_Domain _Instance)
source	FK (Reference)

Table 18 The Domain_Example table

Metaphor_Example	
ID	PK
Instance_Using_Example	FK (Target_Domain_ Instance)
Source	FK (References)

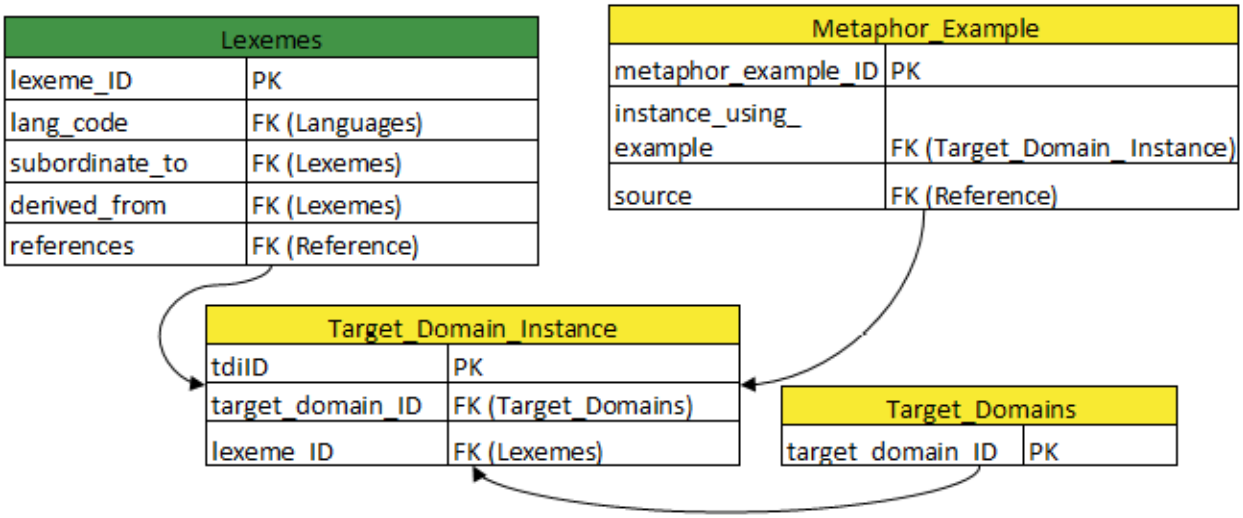


Figure 24. The Lexemes, Target_Domain_Instance, Target_Domains and Metaphor_Example tables.

They both have foreign keys called **instance_Using_Example** linking to **Target_Domain_Instance**, in the case of **Metaphor_Example**, and linking to **Semantic_Domain_Instance**, in the case of **Domain_Example**. Likewise, they are both linked to the **References** table (which will be discussed shortly) through the foreign key **source**. The actual content in these tables are the examples, which can be given in the text field attribute **example_text**, with a translation in the text field attribute **translation**. In this first conceptual design, it was still unclear how to best capture the morphosyntactic behavior of the temperature term under study – an attempt was made to further this by adding an attribute

syntactic_function_of_key_word, in which a closed list (i.e enums) consisting of the values *adjective, verb, noun, participle, adverb* and *other* could be chosen.

An alternative to this approach, i.e creating two identical tables, would be to create three new tables – see alternative B below. Two linking tables, containing nothing but linking foreign keys, which hypothetically might have been called Metaphor_Example_Instance and Domain_Example_Instance, which then link to a single Example table. There are merits to both approaches, but I decided to go with the one outlined above, to cut down on the total number of tables the database would contain.

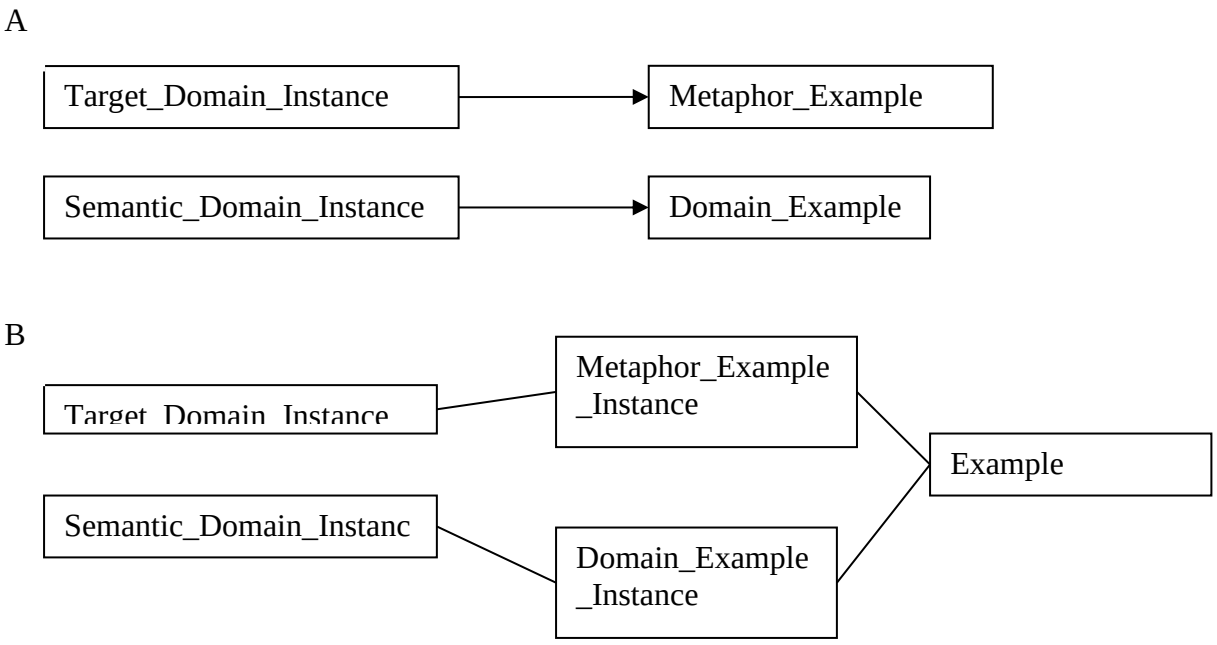


Figure 25. Different approaches to examples.

Chapter 5: Evaluation of the SQL and RDF Attempts

The major advantage of the RDF database described in earlier sections is how easy it is to build many-to-many-relations between any two data instances (such as between any resource and a source, or any resource and a comment). The TDTT is intended for human perusal, not for computer harvesting – but if it were ever to be used for larger scale computational linguistic quantitative research, having the data in RDF, where all types and resources are described at their URIs, would make it much easier for computers to infer semantic connections.

The major drawbacks of RDF solution is that there is too little software support for it, that although there is an impressive standard query language (SPARQL) approved for it, it can only search the data – not update it and that it is as yet relatively unknown (compared to SQL) for most users.

All of these problems might go away in time – but at present they pose big enough hurdles that I decided to abandon the RDF project. The lack of software support means that using it requires major time investment in building interfaces for data management and user interaction – something that is not feasible for small scale research projects to set up, nor to maintain.

The SQL solution has the corresponding weakness that it is difficult to handle sources and comments for any data points. That being said, it is far easier to implement – both in local DBMS such as Microsoft Access, Open Office Base of FileMaker Pro, all three of which enjoy widespread use, and online, through frameworks such as Django, which can (through python)

generate web sites built from SQL databases. An RDF version of Django was started, but never finished[16].

The conceptual design of the TDTT is intended to be superseded by later versions. Once data are added to the database, this will prompt redesigns.

[16] http://code.google.com/p/django-rdf/

Chapter 6: The Physical Design of the SQL Database

This section will briefly present some of the interfaces the database can be used for.

I created a physical design of the SQL logical design described above. I wished to create a simple and easily accessible interface for other researchers, without requiring them to have any special knowledge of databases. This necessitated a powerful graphical interface that was as intuitive as possible. I first decided to use the Microsoft Access database management system to create the tables and input data from the first dozen languages, but ran into problems with poor interoperability between Mac and PC computers.

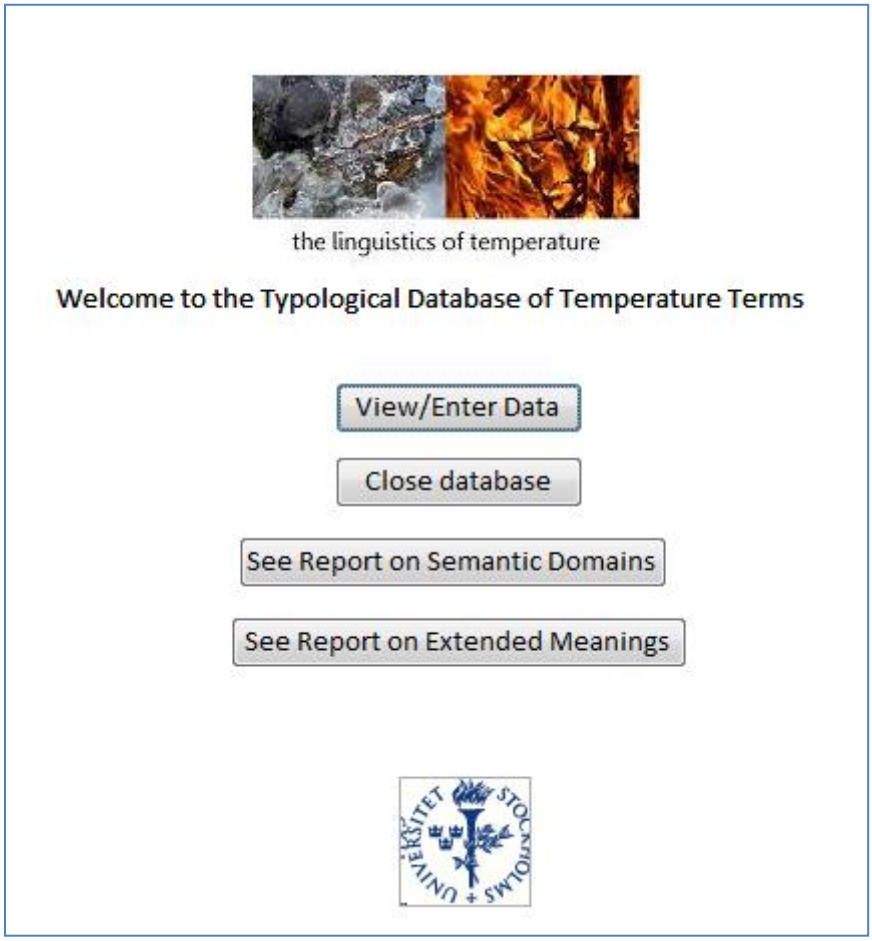


Figure 26. The Microsoft Access interface to the TDTT.

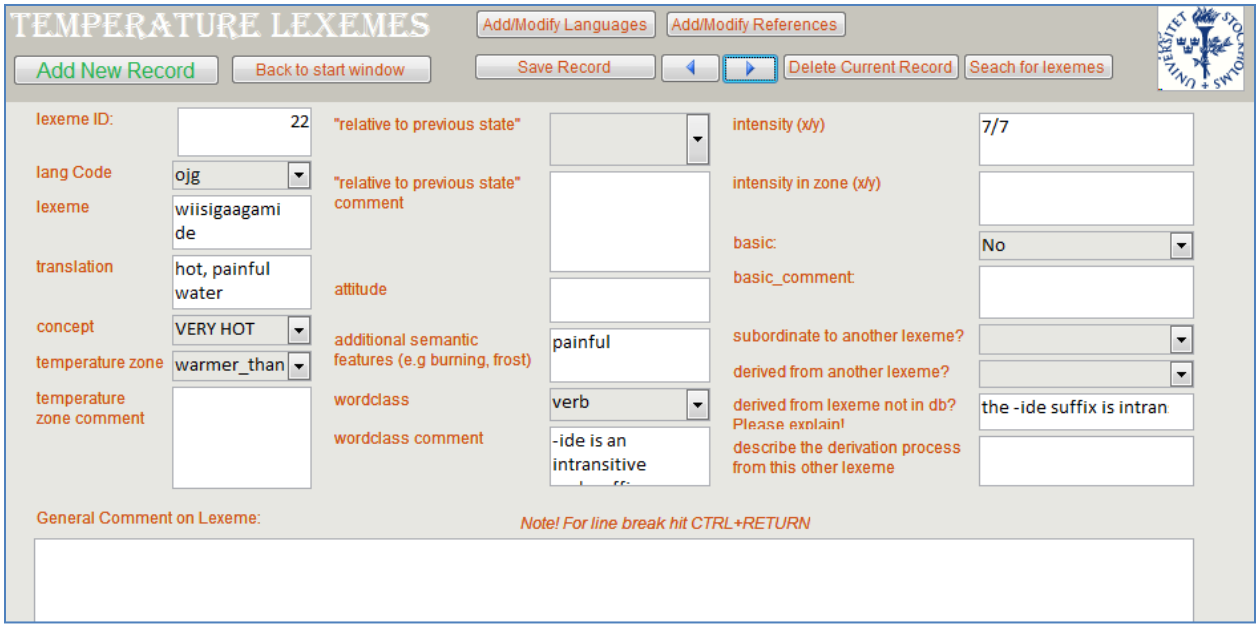


Figure 27. The Microsoft Access interface to the TDTT.

I later migrated the data to the Open Office Base DBMS, which works on both Macs and PCs but found the software too instable and prone to crashes to be useful for extended databases.

The latest management system for the TDTT is a web accessible Django installation (built on a PostGreSQL DBMS and python) which has the virtue of the front end being easy to use for inexperienced users. It has several different ways to view the data – but at the moment, no direct SQL interface for users. Instead, it is possible to generate a report of the database by building a multivariate search query through a visual GUI and specifying the values of a couple of variables. The reason for this was that the full power of SQL queries is rather daunting to inexperienced users – this way, the database becomes an interactive dictionary of temperature terms. In Figure 29 **you** find an excerpt from a search result from the Django online interface.

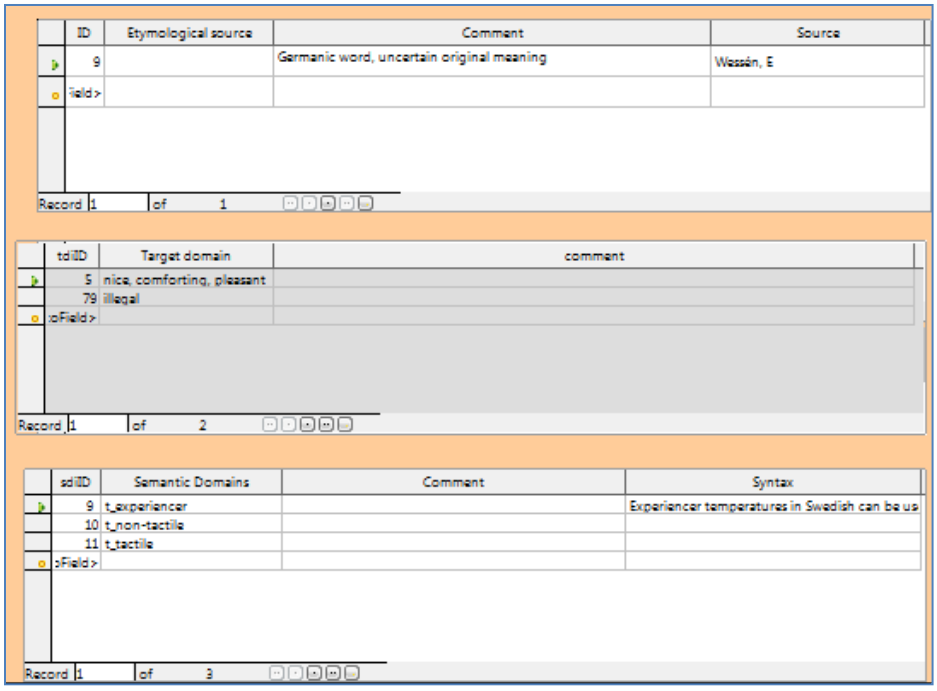


Figure 28. The Open Office Base DBMS and interface of the TDTT.

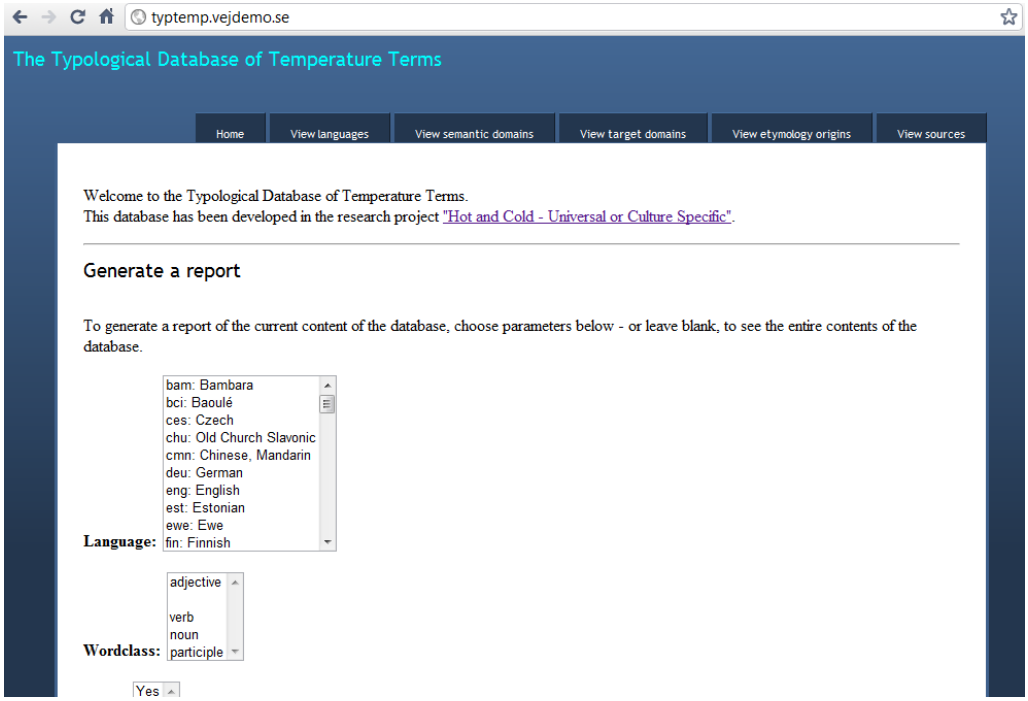


Figure 29. The Django online interface of the TDTT.

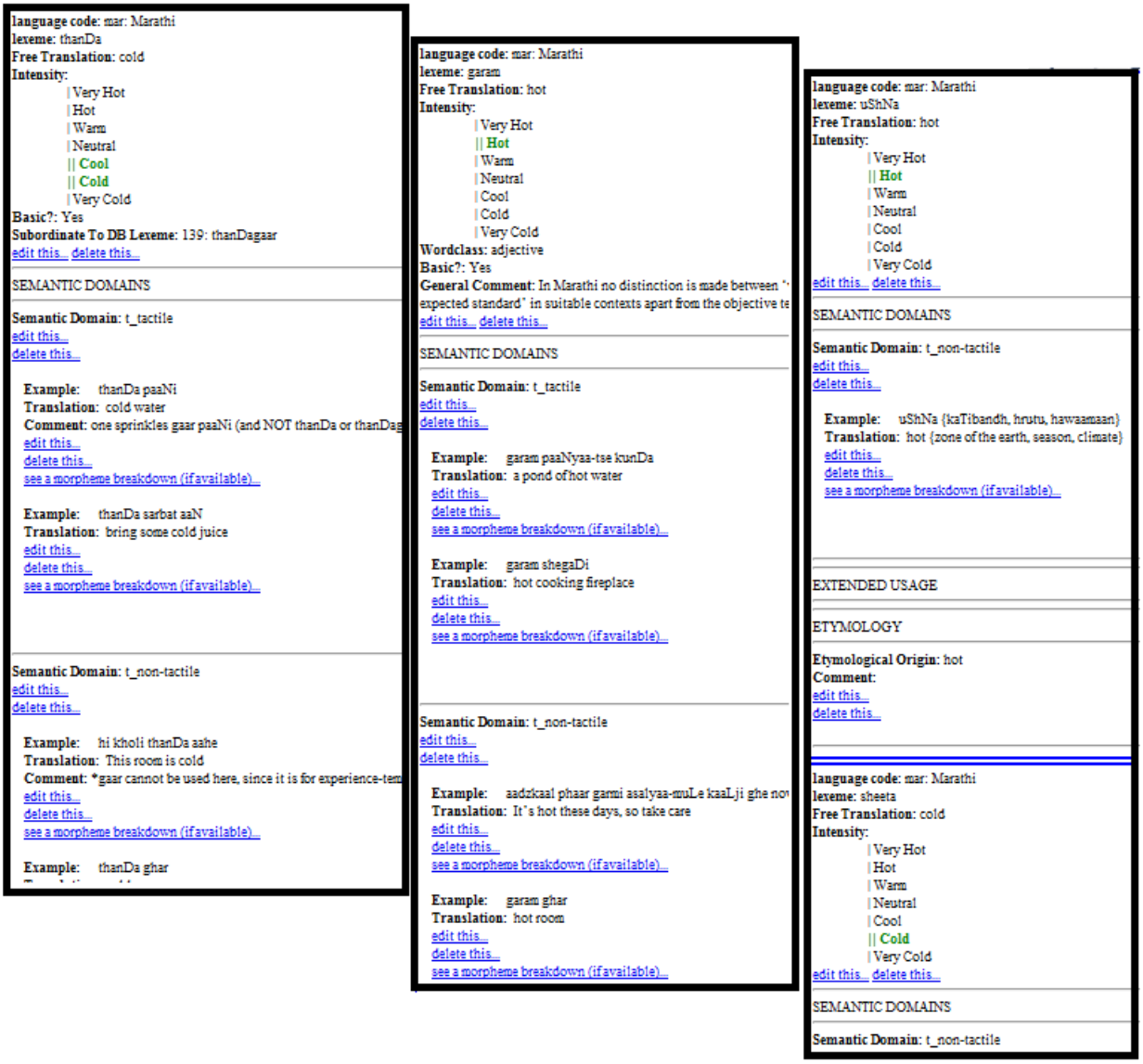


Figure 30. Excerpt of search returns from the Django online interface for Marathi temperature terms.

Chapter 6: Addition of Data to the SQL Database

Table 19. The first batch of languages in the TDTT.

Language Name	Language Family	# of lexemes
Bambara	Niger-Congo	5
Baoulé	Niger-Congo	7
Bora	Tupi-Guarani	3
Chinese, Mandarin	Sino-Tibetan	4
Dutch	West Germannic	15
Eastern Ojibwe	Algonquian	16
Estonian	Finno-Ugric	9
Ewe	Volta-Congo	9
Fe'fe'	Bantu	6
French	Romance	4
German	West Germannic	10
Hausa	West Chadic	3
Hiw	Eastern Malayo-Polynesian	4
Hungarian	Finno-Ugrian	6
Icelandic	North Germannic	9
Igbo	Benue-Congo	2
Italian	Latin	48
Italian, Tuscan	Latin	2
Kalenjin	Nilotic	5
Kilba	Chadic	2
Lemerig	Eastern Malayo-Polynesian	2
Luo	Nilo-Saharan, Nilotic	4
Luyia	Bantu	4
Marathi	Vedic Sanskrit	10
Moba	Gur	4
Mwotlap	Eastern Malayo-Polynesian	1
Ngwo	Bantu	7
Serbian	West South Slavic	8
Swahili	Benue-Congo	4
Swedish	North-Germanic	7
Tagalog	Meso Philippine	3
Ukrainian	Slavic, East	12
Yoruba	Benue-Congo	3

I manually added data from 33 languages to the SQL database. The languages are not a representative sample, and there is a clear over representation of African and European languages. The languages with the largest number of lexemes in the TDTT is Italian (41 terms) and the language with the least number of temperature terms in the TDTT is Mwotlap (one term).

The data come primarily from presentations at the Workshop on Temperature in Language and Cognition held in Stockholm in March 2010, but also from elicitation (Ewe and Italian), personal communications from researchers on temperature terms in their native language (Swedish) and dictionary studies (German).

The scarcity of available data means that I have made no attempts to choose only particular languages, or to limit the kinds of data I have entered in the database. The current outliers in the database when it comes to number of temperature terms is Italian with 48 lexemes and Mwotlap with 1 – once these are removed, the average number of temperature terms in the database per language is six. There is no reason from these numbers to form a conclusion that Italian has a much more nuanced division of the temperature spectrum either – the Italian data contain such very specialized entries as *artico* 'arctic cold; very cold' simply because project contributors have given us a great deal of Italian material. Many of the Italian temperature terms are probably used very seldom and in very particular contexts – but there is no comparative data on their use, so at this stage of the database, I have included them all.

I entered the data between June and August 2010 and then decided to not include any more data until a first analysis of the data structure was done. The goal is to use this analysis to redesign and improve the database, before more data are added.

SQL Redesign

The addition of data into the TDTT led to several redesigns of the database structure. I will discuss these in the following section.

The conceptual design for TDTT shown above can be seen as consisting of a core database with three detachable modules. The core database is the Lexemes table, the Languages table and the Reference table. These three tables hold the central information about the lexeme.

The three additional modules could be named the Etymology module, the Metaphor module and the Semantic Domain module. They are independent of one another. The Etymology module

consists of the table Etymology, which is linked to Lexeme with the linking table Etymology_Instance. The other modules are similar in structure (a Target Domain table linked to the Lexemes table through a linking table, Target_Domain_Instance; a Semantic Domain table linked to the table Lexemes through the linking table Semantic_Domain_Instance) but also have additional "Example modules." The Example modules are easily detachable from the rest of the module (i.e they can be redesigned without affecting the other tables) and consist of Metaphor_Example and Domain_Example tables, respectively.

Once the database was filled with data from a dozen languages, several problems became apparent.

Table 20. Metaphor_Example_Word.

Metaphor_Example_Word	
Name	**Primary Key, Foreign Keys**
example_Word_ID	PK
example_ID	FK
lexeme_Level	
delineated_lexeme_Level	
morpheme_Level	
english_Translation_Level	
comment	

Interlinear Glossed Text and Examples. A problem with this first conceptual design is the lack of support for interlinear glossed text, or morpheme breakdowns, of the example. I decided that the best way to proceed with this was to create an additional table for each of **Metaphor_Example** and **Domain_Example**, called **Metaphor_Example_Word** and **Domain_Example_Word**.

Table 21. Domain_Example_Word.

Domain_Example_Word	
Name	**Primary Key, Foreign Keys**
example_Word_ID	PK
example_ID	FK
lexeme_Level	
delineated_Lexeme_Level	
morpheme_Level	
english_Translation_Level	
comment	

The attributes in these tables were the primary key and ID, namely **example_Word_ID**, the foreign key **example_ID**, which linked the **Metaphor_Example_Word** and **Domain_Example_Word** to the **Metaphor_Example** and **Domain_Example** tables, respectively. Both tables then had four attributes corresponding to four levels of analysis. An example is given below, for the Swedish term *glödhett*.

Table 22. Example of four levels of Lexeme analysis.

lexeme_Level (text)	The lexeme itself	glödhett
delineated_Lexeme_Level (text)	The lexeme delineated into morphemes to be analyzed	glöd-het-t
morpheme_Level (text)	Analysis of the morpheme	glowing-hot-GENDER
english_Translation_Level (text)	Translation of the morpheme	glowing hot

Another problem with the first conceptual design is the fact that there is a single attribute, **translation**, for metaphorical examples: it might be better if there had been room for both a literal and a metaphorical interpretation. I decided to add these two attributes,

literal_interpretation and **metaphorical_interpretation** to the database and remove the attribute **translation**.

Sources. The main advantage that the RDF solution has over the SQL solution is the treatment of sources and comments. In the SQL solution comments are linked to the attributes they are commenting only by their names (as when the text field attribute **Basicness_comment** is "linked" to the Boolean attribute **Basicess**). Had the database been intended for digital data harvesting of some kind, where the data structure needs to be interpretable by computers, this would have been a major problem. Since that is not the primary purpose of the TDTT, the system works.

Since most data in a given row in a table of the database presumably comes from the same source, the database only allows one **source** attribute per table. This will be a problem when and if individual attributes in the same row have different sources – but that has not happened in the data added to the database so far. A related problem is that in the first conceptual design, not all tables were linked to the Reference table: in fact, only Lexemes, Metaphor Example, Domain Example and Etymological Instance were linked. The reason for this was that I considered these tables to be the ones most likely to need citation information. However, once data were added to the database, I discovered this to be a faulty supposition and I remedied this by linking all tables, with the exception of the **Metaphor_Example_Word** and **Domain_Example_Word** tables, to the **Reference** table.

Basicness status. In the first conceptual and logical design I attempted to gather the information about a lexemes basicness status through two attributes in the Lexeme table: A Boolean attribute **basic** and a text field attribute **basic_comment**.

As data were added to the database, it became apparent that there occasionally existed more detailed information that it would be good to store in the database on this issue. Therefore I added several attributes to the Lexeme table: Frequency_Of_Use, (is the lexeme very frequent and generally known?), **Frequency_Comment** (further comments on the frequency of the lexeme), **Native**, (is the lexeme of native origin?), **Native_Comment** (further comments on origin), Simple_**MorphoSyntax** (is the L. morphosyntactically (relatively) simple and non-compositional?) and **Simple_MorphoSyntax_Comment** (further comments on Morphosyntax).

Intensity levels. In the first analysis, it was decided that the temperature intensity level of each lexeme would be indicated by two attributes. The attribute **intensity_total** consisted of a text field with two numbers: X/Y, where Y were the total number of temperature intensity levels in the language and X was the level that the term in question was at. The attribute **intensity_in_zone** was also a text field with two numbers: Z/W, where W was the total number of intensity levels in the warm zone, the neutral zone of the cold zone for the language and Z was the intensity level that the lexeme in question reflected.

While this model would be very useful if the full information was available for each language added to the database, it proves difficult when only partial information about the language temperature term system is available.

Instead, I searched for an alternative way of documenting intensity of temperature. The available literature on cross-cultural temperature research often uses a seven level intensity scale where one is the coldest and seven is the hottest. This has proved a useful tool for engineers working on climate control. There are two seven level scales in general use, the Bedford scale (Bedford 1936) and the ASHRAE scale[17] - the differences between them are irrelevant for this thesis. I decided to create seven new attributes for the **Lexemes** table: **concept_VeryHot, concept_Hot, concept_Warm, concept_Neutral, concept_Cool, concept_Cold** and **concept_VeryCold**. Each of these attributes are Boolean: a lexeme either expresses a particular level, or it does not. The person inputting data has to imagine a scale of seven levels of temperature intensity and choose which level or levels a certain temperature term reflects. I believe this will yield interesting cross-linguistic data and also enable fruitful cross-linguistic comparisons.

Morphosyntactic change and semantic domains. Once data were added to the database, Maria Koptjevskaja Tamm (p.c) pointed out that the first design did not do enough to document the interaction between semantic and morphosyntactic change. There were only text field attributes where the morphosyntactic patterns associated with particular semantic domains could be entered – something that did not allow for easy queries or cross-linguistic comparison. It would be interesting to know the morphosyntactic standard pattern for both attributive and predicative use of the lexemes, as well as which morphosyntactic changes were made for attenuation or intensification of the lexemes. I decided to do some major changes to the Semantic_Domain_Instance table.

[17] The American Society of Heating, Refrigerating and Air-Conditioning Engineers, http://www.ashrae.org/

The predicative and attributive uses of a lexeme often have very different morphosyntax – and it is also possible that a particular lexeme can only be used in either an attributive or a predicative fashion, or that they are more frequently used in an attributive or a predicative fashion. In the first design, there was no way to discuss the attributive uses and the predicative uses separately. I decided to create two new attribute in the **Semantic_Domain_Instance** table, labeled **attributive_use** and **predicative_use**. In addition, I created the text field attributes **attributive_comments, attributive_morphosyntax, predicative_comments** and **predicative_morphosyntax**.

I also created two text fields, **attenuation** and **intensification** where users could enter data about the standard attenuation and intensification methods for the lexeme.

While these decisions will make it easier to separate different kinds of information about the morphosyntactic behavior of the lexeme in predicative and attributive use, it is still hard to do cross-linguistic comparison between large amount of data, when the data are placed in text fields. However, we simply do not know enough about cross-linguistic attenuation and intensification techniques of temperature terms to create anything but text fields at the moment. Hopefully, at a later date, the information given in the text field attributes can be used for further re-designs of the database.

An even larger redesign was needed for the table Semantic_Domains. Previously, I had used a simple taxonomy, which could be infinitely extended when new restrictions on the uses of temperature terms became prevalent in languages. Initially, the taxonomy contained four entries – which were stored as rows in the Semantic_Domain table, as can be seen in Table 23.

Table 23. Semantic Domains.

Semantic_Domain		
ID	**Label**	**Semantic_Domain_Comment**
1	temperature	Level 1 node. The lexeme can be used for all kinds of temperature expressions
2	t_tactile	Level 2 node. The lexeme can be used for tactile temperature expressions
3	t-non-tactile	Level 2 node. The lexeme can be used for non-tactile temperature expressions
4	t-experiencer	Level 2 node. The lexeme can be used for experiencer temperature expressions

As data were added to the database, more semantic domains were discovered and more rows were added to the Semantic_Domain table. Every time a lexeme had some sort of restriction on its use – for instance, that it could only be used to describe the temperature of liquids – a new row was added and a new entry in the ontology was made. The final list of semantic domains after the first design read as follows in Table 24.

Table 24. Semantic Domains.

Semantic_Domain		
ID	**Label**	**Semantic_Domain_Comment**
1	temperature	Level 1 node. The lexeme can be used for all kinds of temperature expressions
2	t_tactile	Level 2 node. The lexeme can be used for tactile temperature expressions
3	t-non-tactile	Level 2 node. The lexeme can be used for non-tactile temperature expressions
4	t_experiencer	Level 2 node. The lexeme can be used for experiencer temperature expressions
5	t_tactile_water	Level 3 node. The lexeme can be used for water
6	t_tactile_bodyparts	Level 3 node. The lexeme can be used for body parts
7	t_tactile_surfaces	Level 3 node. The lexeme can be used for surfaces
8	t_tactile_snowIce	Level 3 node. The lexeme can be used for snow and/or ice
9	t_tactile_metal	Level 3 node. The lexeme can be used for metal
10	t_tactile_food	Level 3 node. The lexeme can be used for food
11	t-non-tactile_sun	Level 3 node. The lexeme can be used for temperature caused by the sun
12	t-non-tactile_fire	Level 3 node. The lexeme can be used for temperature caused by fire

13	t-non-tactile_weather	Level 3 node. The lexeme can be used for temperature caused by weather
14	t-non-tactile_ weather_wind	Level 4 node. The lexeme can be used for temperature caused by wind
15	t-non-tactile_clothes	Level 3 node. The lexeme can be used for temperature caused by clothes (“this shirt is warm”)
16	t-non-tactile_periods	Level 3 node. The lexeme can be used for temperature during a particular time period.

While the use of this taxonomy had originally seemed like the best way to order the data, once the data were added it became apparent (Maria Koptjevskaja Tamm, pc.) that the database was imposing a less than beneficial structure on the data by connecting the three semantic domains (tactile, non-tactile and experiencer type temperatures) that are often given different morphosyntactic and lexical treatment in languages, to restrictions in the kinds of entities the temperature terms could modify. For instance, clothes that cause the wearer to feel warm can themselves thereby be called ‘warm’ in several languages. But in certain languages, these clothes expressions are very similar to tactile temperature expressions – and in others, they are similar to non-tactile temperature expressions. But the first taxonomy makes it necessary to determine if clothing temperature should be a tactile or non-tactile concept.

I decided that the best course of action would be to split up the taxonomy. Each lexeme can have many different semantic domain instances and each semantic domain instance can have many different restrictions on the kinds of entities it can modify. I thus created a table called **Entity**, which was connected to the **Semantic_Domain_Instance** table through the **Entity_Instance** linking table. The **Entity** table has the attributes **entity_ID** (Primary key), **entity_Name** (text field) and **entity_Comment** (text field), as well as a foreign key attribute **source**, which connected it to the **Reference** table.

Any lexeme which had previously been connected to the semantic domain non-tactile_clothes will now be connected to the semantic domain non-tactile and the entity clothes.

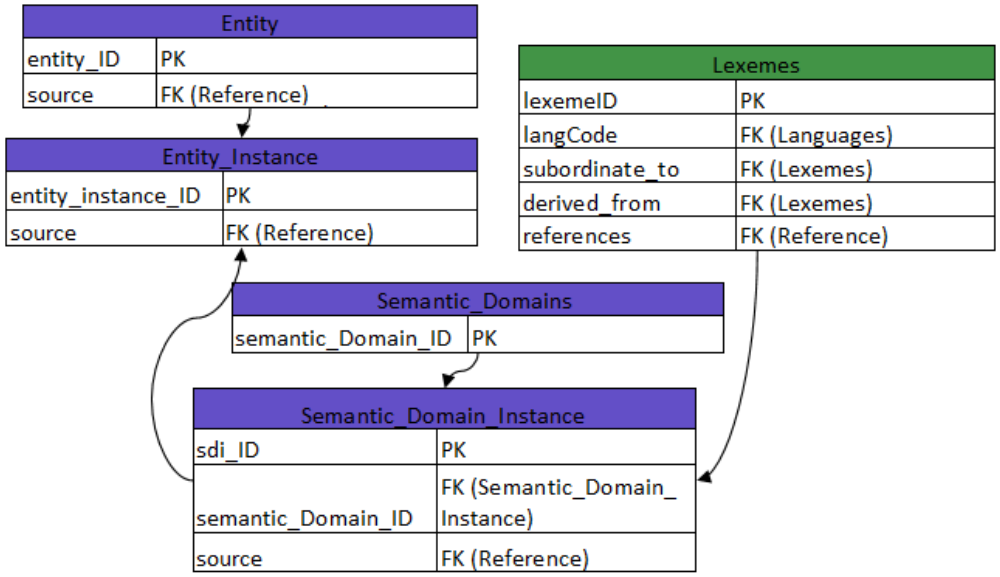
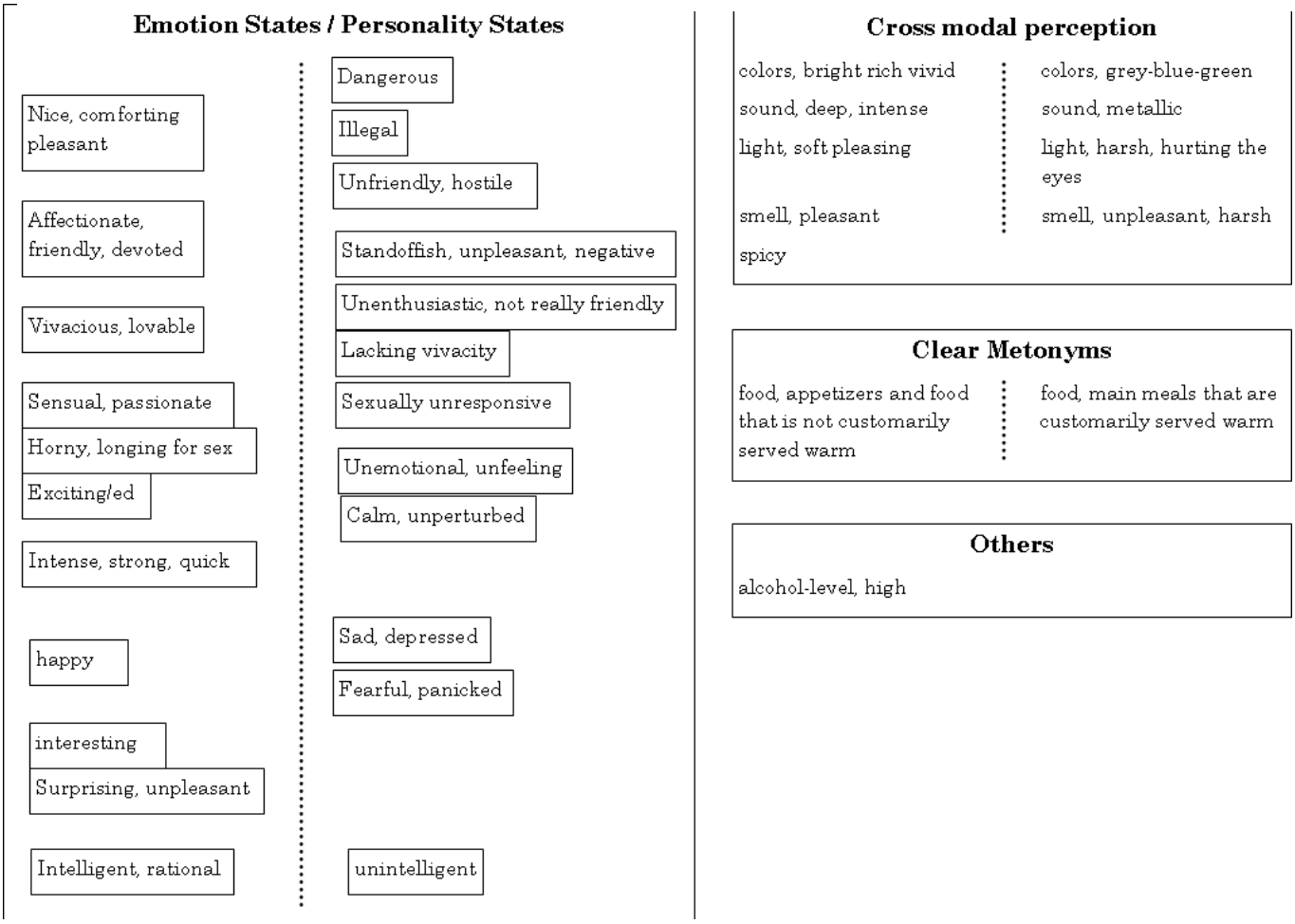


Figure 31. The Entity, Entity_Instance, Semantic_Domain_Instance, Semantic_Domain and Lexemes tables.

Extended uses. The first design of the metaphor module of the database – i.e the tables Target_Domain_Instances, Target_Domains, and Metaphor_Example let the user associate a particular semantically extended use of a temperature term with a Target_Domain. It was possible to either chose an existing Target_Domain or create a new one. Once the first batch of data were added, no less than 42 different Target_Domains had been added. These are listed in Table 25, where they have been grouped in three main groups (emotional states/personality states, cross-modal perception, clear metonyms and others). The first three of these groups contain several concepts that are antonyms – antonym pairs are separated by a dashed line.

Table 25. The target domains for temperature metaphors from the TDTT.



While the kinds of target domains that appear are interesting from a research point of view, it is also clear that some of the target domains are semantically closer to each other than others – for instance, the target domains "Standoffish, unpleasant, negative" and "surprising, unpleasant" share the characterization "unpleasant." It would be good if this is also reflected in the database.

Rather than trying to characterize every distinctive figurative use of a lexeme with a single target domain, I decided to break the target domains into the smallest units, and then attach several of these smaller units to each figurative use. The change is illustrated Figure 32 below, where the boxes represent target domains and the circles lexemes.

In the relational database, I created a new table called **Tag_Instance** and inserted as a link between **Target_Domain_Instance** and **Target_Domain**. This lets each **Target_Domain_Instance** have several different tags.

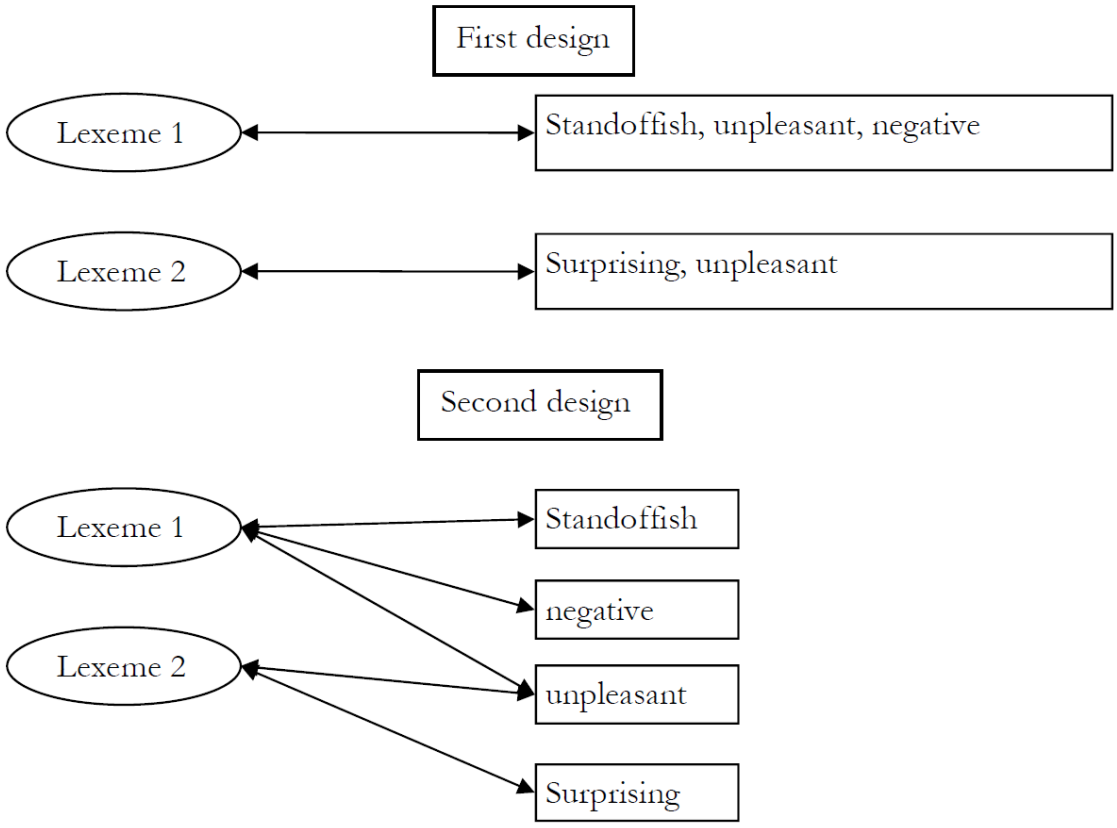


Figure 32. The first and second design of target domain handling in the TDTT.

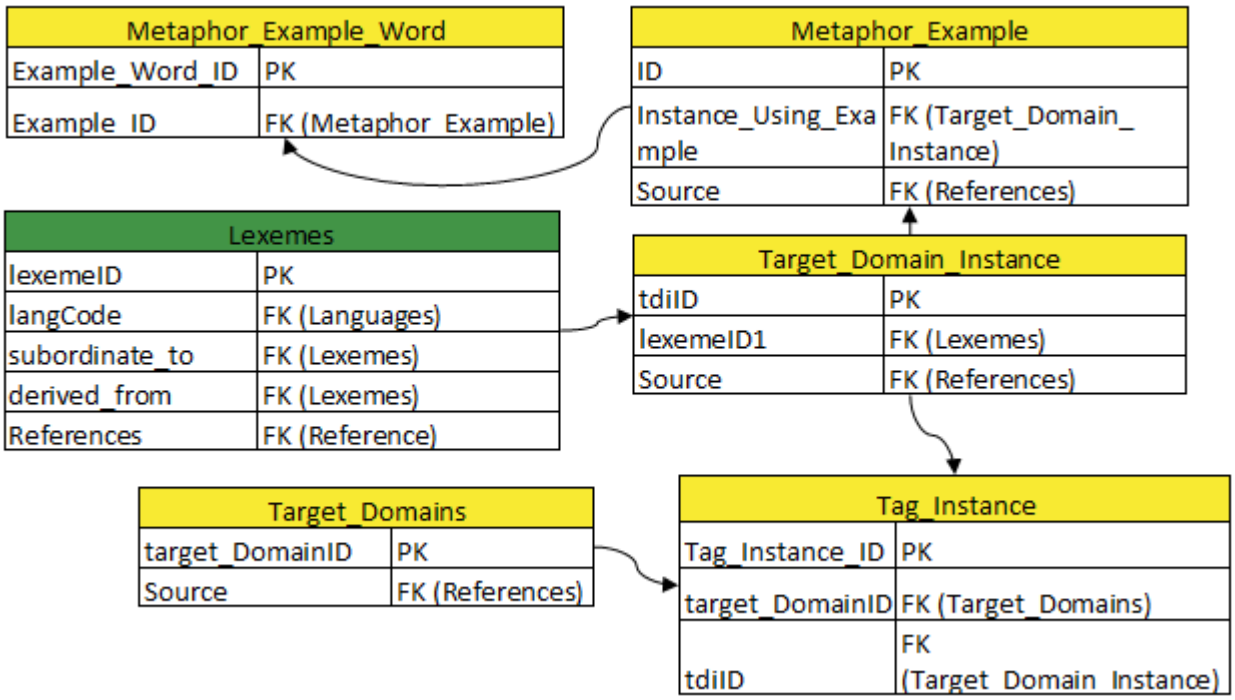


Figure 33. The detailed workings of the Lexemes, Target Domains, Tags, Meraphor example etc. tables.

Chapter 7: Summary

The database is foremost an electronic dictionary. Its intended users are researchers into the lexical semantics of temperature terms. The online Django interface lets users perform a limited set of queries of the data on language, intensity level, metaphor domain, target domain and etymology.

The SQL database, which will be downloadable from the interface, permits more advanced users access to all the database tables and will let them find cross-linguistic similarities in the way different languages treat the temperature domain.

Apart from the practical usefulness of the finished database, this project has also led to several conclusions about design of small scale typological research databases.

In lieu of better, easily available and free interface solutions for RDF databases, SQL must remain the preferred alternative for small scale research projects.

For research into lexical semantics, a feature bag approach is preferable to using a taxonomy tree.

Most users of this kind of database will not need access to the full power of a query language – instead it is vital to supply them with good views where they can peruse the material like an interactive dictionary. Manual scrutiny of data is just as important as the ability to do quantitative searches over it.

Finally, the database structure of this database can be re-used for other, similar research projects into lexical semantics. Small projects need not use all the modules (database tables) developed for the temperature project, but can pick and choose those that best fit their intended data.

References

Abed Al-Haq, Fawwaz, and Ahmad El-Sharif. 2008. ‘A Comparative Study for the Metaphors Use in Happiness and Anger in English and Arabic’. *US-China Foreign Language* 6 (11).

Baker, Collin, Charles Fillmore, and Beau Cronin. 2003. ‘The Structure of the Framenet Database’. *International Journal of Lexicography* 16 (3) (September 1): 281–296.

Berlin, Brent, and Paul Kay. 1969. *Basic Color Terms Their Universality and Evolution*. Berkeley: University of California Press.

Boas, Hans, ed. 2009. *Multilingual FrameNets in Computational Lexicography : Methods and Applications*. New York NY: Mouton de Gruyter.

De Luca, Ernesto William, Martin Eul, and Andreas Nürnberger. 2007. ‘Converting EuroWordNet in OWL and Extending It with Domain Ontologies’. In *Proceedings of the Workshop on Lexical-Semantic and Ontological Resources. In Conjunction with the GLDV-Frühjahrstagung (GLDV 2007)*. Tübingen.

De Luca, Ernesto William, and Birte Lönneker-Rodman. 2008. ‘Integrating Metaphor Information into RDF/OWL EuroWordNet’. *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08), Marrakech, Morocco*.

De Luca, Ernesto William, and Andreas Nürnberger. 2006. ‘LexiRes: A Tool for Exploring and Restructuring EuroWordNet for Information Retrieval’. In *Proceedings of the Workshop on Text-Based Information Retrieval (TIR-06). In Conjunction with the 17th European Conference on Artificial Intelligence (ECAI’06)*. Riva del Garda, Italy.

Everaert, Martin, Simon Musgrave, and Alexis Dimitriadis. 2009. *The Use of Databases in Cross-Linguistic Studies*. Berlin ;;New York: Mouton de Gruyter.

Fellbaum, Christiane. 1999. *WordNet an Electronic Lexical Database*. 2nd print. Cambridge Mass: MIT Press.

Fillmore, Charles. 1992. 'Towards a Frame-Based Lexicon: The Semantics of RISK and Its Neighbors'. In *Frames, Fields, and Contrasts : New Essays in Semantic and Lexical Organization*, edited by Adrienne Lehrer and E Kittay, 75–102. Hillsdale N.J.: L. Erlbaum Associates.

Firsching, Henrike. 2010. 'Temperature Terms in 14 African Languages'.

Francois, Alexandre. forthcoming. 'Temperature Terms in Northern Vanuatu'.

Horák, Aleš, and Pavel Smrž. 2004. 'VisDic - Wordnet Browsing and Editing Tool'. In *Proceedings of the Second International Conference of the Global WordNet Association (GWC 2004)*, 136–141. Brno, Czech Republic.

Koptjevskaja-Tamm, Maria. 2007. 'Guidelines for Collecting Linguistic Expressions for Temperature Concepts: Version 1 (December 2007)'.

Koptjevskaja-Tamm, Maria, and Ekaterina Rakhilina. 2006. '"Some like It Hot": On the Semantics of Temperature Adjectives in Russian and Swedish'. *Sprachtypologie Und Universalienforschung : STUF.* 59 (3): 253.

Lakoff, George, Jane Espenson, and Alan Schwartz. 1991. 'Master Metaphor List (Second Draft Copy)'.

Lakoff, George, and Mark Johnson. 1980. *Metaphors We Live by*. Chicago: University of Chicago Press.

Lami, Irene. 'Italian Temperature Term Research Notes'.

Lönneker-Rodman, Birte. 2008. ‘The Hamburg Metaphor Database Project: Issues in Resource Creation’. *Language Resources and Evaluation* 42 (3) (October): 293–318. doi:10.1007/s10579-008-9073-9.

Narayanan, S, C Baker, C Fillmore, and M. R.L Petruck. 2003. ‘FrameNet Meets the Semantic Web: Lexical Semantics for the Web’. *Lecture Notes in Computer Science.* (2870): 771–787.

Petruck, M. R.L. 1996. ‘Frame Semantics’. In *Handbook of Pragmatics,* edited by Jef Verschueren, Jan-Ola Östman, Jan Blommaert, and Chris Bulcaen, 1–13.

Princeton Wordnet. 2010. ‘Princeton WordNet Search’. Accessed October 14. http://wordnetweb.princeton.edu/perl/webwn.

Rodríguez, Horacio, Salvador Climent, Piek Vossen, Laura Bloksma, Wim Peters, Antonietta Alonge, Francesca Bertagna, and Adriana Roventini. 1998. ‘The Top-Down Strategy for Building EuroWordNet: Vocabulary Coverage, Base Concepts and Top Ontology’. *Computers and the Humanities* 32 (2/3) (January 1): 117.

Ruppenhofer, J., M. Ellsworth, M. R.L Petruck, C. R Johnson, and J. Scheffczyk. 2010. ‘Framenet II: Extended Theory and Practice’. http://framenet.icsi.berkeley.edu/book/book.pdf.

Schmidt, Thomas. 2009. ‘The Kicktionary - A Multilingual Lexical Resource of Football Language / Thomas Schmidt’. In *Multilingual FrameNets in Computational Lexicography : Methods and Applications*, edited by Hans Boas, 101–134. New York NY: Mouton de Gruyter.

Sutrop, Umas. 1998. ‘Basic Temperature Terms and Subjective Temperature Scale’. *LEXICOLOGY -BERLIN-* 4 (1): 60–104.

Vejdemo, Susanne, and Sigi Vandewinkel. submitted. ‘Extended Uses of Temperature Terms across Languages’. In *Lexicotypological Approaches to Semantic Shifts and Motivation Patterns in the Lexicon*, edited by Maria Koptjevskaja-Tamm and Päivi Juvonen. Berlin: De Gruyter Mouton.

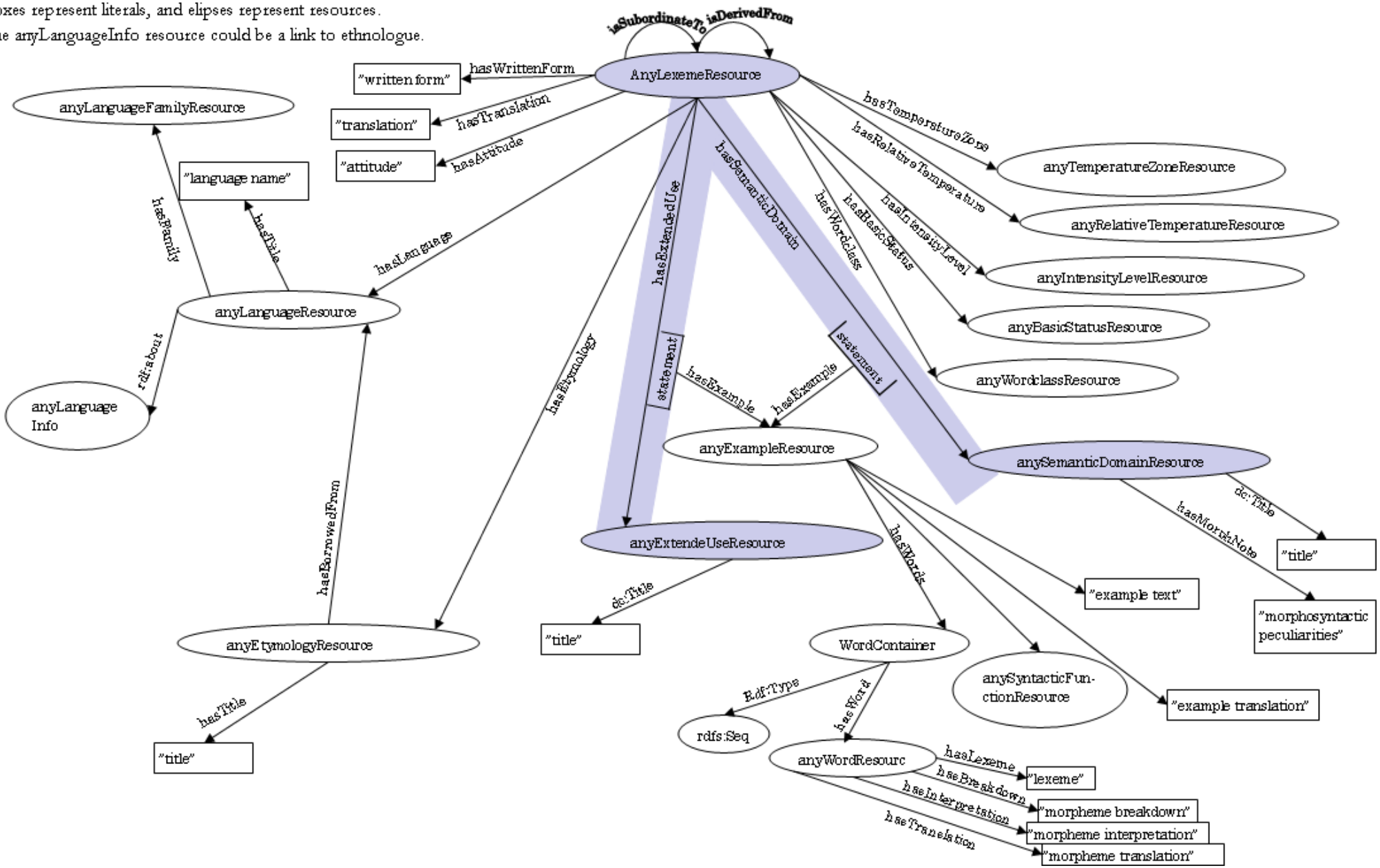
Viberg, Åke. 2001. ‘Polysemy and Disambiguation Cues across Languages. The Case of Swedish Få and English Get’. In *Lexis in Contrast : Corpus-Based Approaches.*, edited by Bengt Altenberg, 119–150. Amsterdam: John Benjamins Publishing Company.

Vossen, Piek. 1998. *EuroWordNet : A Multilingual Database with Lexical Semantic Networks*. Dordrecht [The Netherlands] ;;Boston: Kluwer Academic.

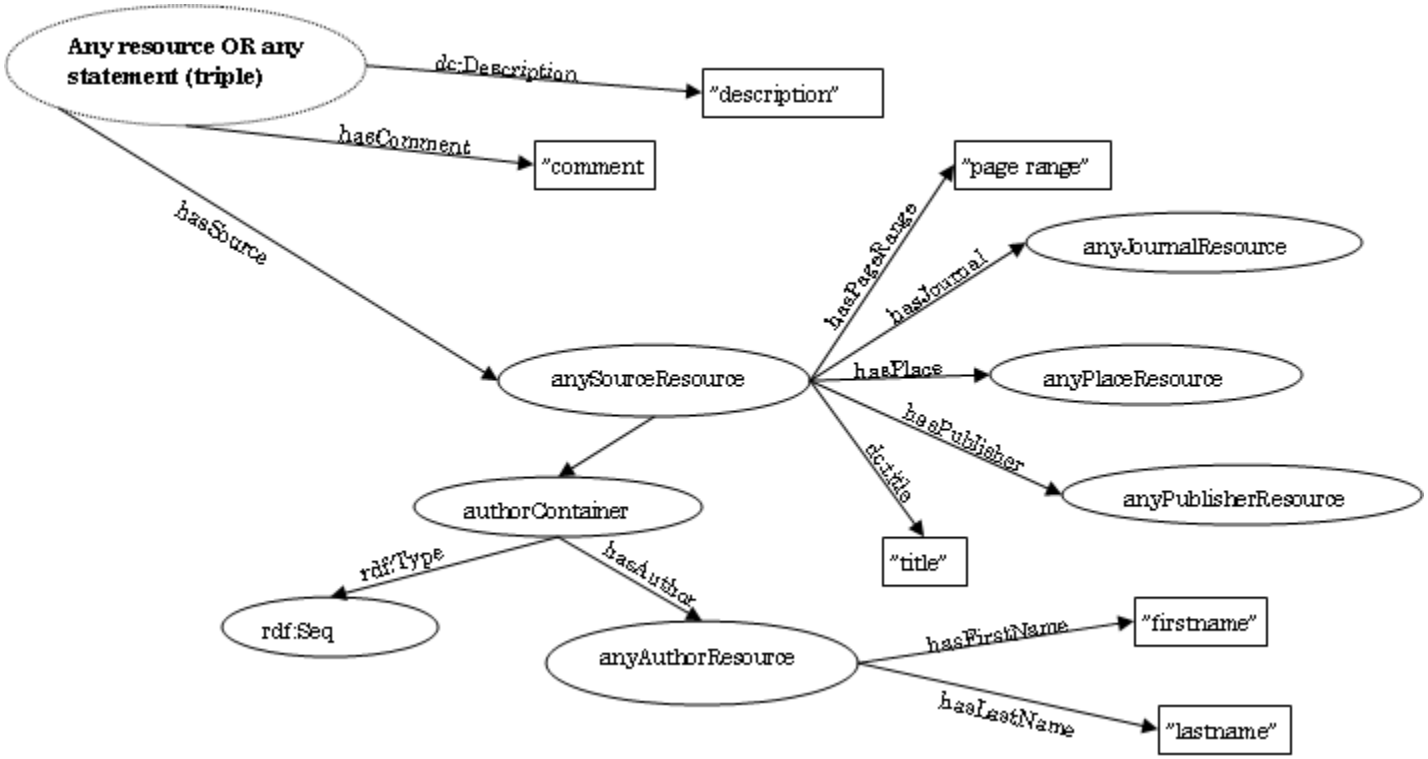
Zalizniak, Anna. 2008. ‘A Catalogue of Semantic Shifts: Towards a Typology of Semantic Derivation’. In *From Polysemy to Semantic Change : Towards a Typology of Lexical Semantic Associations*, edited by Martine Vanhove, 217–232. Amsterdam ;;Philadelphia: John Benjamins Pub.

This is a conceptual model, and does not display full URIs, nor most properties of resources.
Prefixes:
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
no prefix: in logical model, a system for URIs should be established.
Note: the hasExample predicate has as its subject a full statement (i.e a full triple).
Boxes represent literals, and elipses represent resources.
The anyLanguageInfo resource could be a link to ethnologue.

Appendix 1: The RDF conceptual model p 1/2



Appendix 1: The RDF conceptual model p 2/2



Appendix 2: The SQL conceptual model

This conceptual model represents 1-to-many relationships by arrows (1 → many) and 1-to-1 relationships by lines (1 – 1).