Improving Access to Text
IMPACT

D-EE2.1 IMPACT LEXICON DATABASE STRUCTURE, v3.0 | EE2

Start: Month 1 Due: Month 6 Actual: Month 10, 26, 38

Internal / external	Internal		
Activity type	RTD		
Participant number	10	5	12
Participant short name	INL	DNB	LMU
Estimated person months per participant for this deliverable	6	2	2
Dissemination level	CO Confidential only for members of the consortium (including the Commission Services)		

Document history

Revisions

Version	Status	Author	Date	Changes
0.1	Draft	INL & CIS (LMU) IMPACT teams	1 July 2008	Created
0.2	Draft	Joachim Korb, Jeanna Nikolov & Sven Schlarb (ONB)	5 August 2008	Minor changes
1.0	Final	INL & CIS (LMU) IMPACT teams	30 September 2008	Incorporating changes
2.0	Final	"	22 February 2010	LMF format added
3.0	Final	"	1 March 2011	Added structure for multi-word NE's + section 3.14

Approvals

Version	Date of approval	Name	Role in project	Signature
1.0	23 October 2008	Max Kaiser, Hildelies Balk	SP EE leader, Project Director	OK
2.0	1 March 2010	"	"	OK
2.0	4 March 2011	"	"	OK

Distribution

Version	Date of sending	Name	Role in project
0.1	1 July 2008	Max Kaiser, Hildelies Balk, Klaus Schulz, Uli Reffle, Barbara Pfeifer	SP EE leader, General Project Manager, WP members for EE3
0.2	5 August 2008	Katrien Depuydt	WP leader for EE3
1.0	23 October 2008	Hildelies Balk	General Project Manager
		All (Sharepoint)	All project members
2.0	28 February 2010	Max Kaiser, Hildelies Balk	SP EE leader, Project Director
2.0	1 March 2010	Liina Munari	EC Project Office
3.0	1 March 2011	Max Kaiser, Hildelies Balk	SP EE leader, Project Director
3.0	7 March 2011	Liina Munari	EC Project Office

KB IMPACT is supported by the European Community under the FP7 ICT Work Programme. The project is coordinated by the National Library of the Netherlands

IMPACT Lexicon database structure

1. Introduction

IMPACT lexica are computational lexica which will be used in two ways: in OCR to enhance word recognition, and in Enrichment, to enable variation-independent searches. The core database objects are word forms, lemmata and documents[2]. All other objects define some kind of relation between these.

In order to enable the OCR's spellchecking mechanism to assess the plausibility of the occurrence of a word in a certain text, it is not sufficient to convert existing lexica and dictionaries into a large word list. We also need to

1. Keep track of the sources from which we took the words (Lexical Source, cf. section 6)
2. List the actually encountered words in the language and record occurrences in actual texts, with frequency information (attestation, cf. section 2.5)
3. Record in what kind of texts these words occur (document properties, cf. section 4)

It is impossible to extract all possible word forms from the limited amount of available reliably transcribed historical text. Hence, we need mechanisms to extend the lexicon and to be able to assess the plausibility of "*hypothetical*" words without previous attestations, i.e. words we have not seen before. Supporting data for these mechanisms have to be present in the database, such as:

1. Unknown inflected forms of lemmata which already are in the database can be dealt with by means of the automatic expansion from the lemma to the full paradigm of word forms (paradigmatic expansion, the database information for this purpose is discussed section 5)
2. New spellings of known words can be dealt with by developing a good model of the spelling conventions of the period at hand (cf. section 5.1 for the storage of orthographic variant patterns)
3. Previously unseen compounds can be dealt with by means of a good model of word formation (cf. section 3.10 for the associated database information)

In order to effectuate word searches without having to worry about inflection and variation of wordforms, Enrichment will use "modern lemmata" as variation-independent retrieval keys for the full spectrum of inflectional and orthographical variation.

The database structure is most conveniently discussed by dividing the information into a few main blocks:

1. Information attached to word forms, either unlabeled (i.e. not yet lemmatized or labeled with Part of Speech) or labeled (i.e. with lemma and possibly PoS), cf. section 2.
2. Information attached to lemmata (section 3)
3. Information about documents, parts of documents, document collections (section 4)
4. Auxiliary information needed for expansion and for plausibility-of-new-words prediction (section 5)
5. Lexical Source (section 6)

[2] "Document" is understood here as a sequence of words, together with the document metadata (section 4)

Status of information: external or internal, optional or mandatory
Part of the lexicon database information is intended to be delivered to other work packages, other information is present because it is useful in the lexicon building process.
We specify which information is really a deliverable part of the EE3 output.

A survey of the database fields can be found in the Database scheme (Appendix A). Appendix B briefly touches on the lexicon API in development. An XML interchange format is proposed in Appendix C.

2. Information attached to word forms

There are two distinct objects in the database on the word form level: unlabeled wordforms (i.e. without linguistic information attached to them) and labeled ones (i.e. labeled with lemma and part of speech)

2.1. Database information for unlabeled word forms

Unlabeled word forms may be used in OCR. They only need to be attested in texts. Attestation information is the only kind of information we link to unlabeled forms. (cf section 2.5, attestation)

Status of attestation information for unlabeled word forms: Mandatory, external (use in TR5)

2.2. Part of speech

Each labeled word form is linked to one or several lemmata and assigned a Part of Speech (part_of_speech) label. This grammatical tag[3] is more specific than the one assigned to the lemma (cf. 3.3), as it may include information about inflection, tense, number.
It is not yet clear how much detail needs to be included in IMPACT lexica. We might accept a certain level of underspecification, because clearly, the distinction between formally identical positions in the paradigm is beyond the scope of IMPACT. So instead of tagging 'loopt' as a second or third person singular (which means a lot of effort has to be put in disambiguation), we may mark it simply as a finite verb ending with *–t*.[4]

Status of this information: Part of speech is not externally required, but hardly dispensable, as the relation between a lemma and its inflected forms cannot be defined without it.

2.3. Lemma

Field content: the ID of the relevant lemma object.

Status: mandatory

2.4. Paradigmatic relation between word form and lemma

It is essential that the lexicon explicitates the paradigmatic relations between lemmata and their word forms.

[3] To refer to this grammatical tag as "Part of Speech" is an abuse of terminology.
[4] Cf Bień (2004) for a discussion of this distinction between a "morphological word" (finite verb ending with *–t*) and a "morphosyntactic word" (third person singular).

Inflected forms

This information is not about the formal structure of the inflected form, but merely serves to interlink lemmata and inflected forms. This link is stored in objects of type AnalyzedWordform, which have a PoS property and link to the lemma on the one hand and to the wordform on the other hand. See Figure 1 for a representation of the database structure and Table 1 for an example.

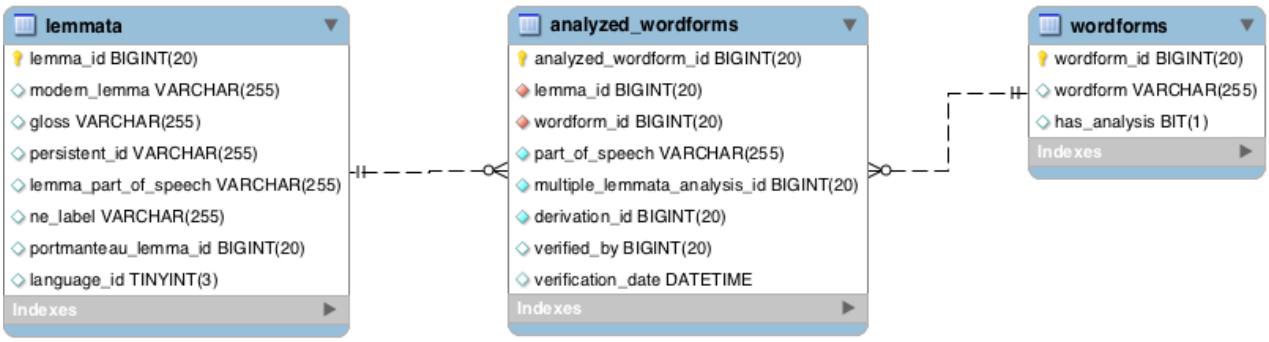


Figure 1: database model[5] for analyzed word forms[6]

Table 1.

Table lemmata

lemma_id	modern_lemma	lemma_part_of_speech
L1	Marcher	VRB

Table analyzed_wordforms

analyzed_wordform_id	part_of_speech	lemma_id	wordform_id
A1	VRB(fin,-erons)	L1	W00001

Table wordforms

wordform_id	wordform
W00001	marcherons

Clitic combinations[7]

[5] The diagrams in this document are in "Crow's Foot" notation. They have been generated from the database by Mysql Workbench 5.0.22. As a result, all relations are annotated as being of the 1:m type with both referencing and referenced table marked as mandatory. This means that some of the logical constraints are not accurately reflected in the diagrams.

[6] The structure changed with respect to the previous version. Instead of just a flat sequence, hierarchy is now possible. It is unlikely that we will use this very much, but we had to incorporate the possibility of having at least two levels because of clitic combinations occurring inside multiword expressions.

[7] We use the term '*clitic combination*' to refer to word forms like dutch '*neemtse*', which is a combination of a finite verb form (neemt = german nimmt) and an unstressed, phonetically reduced pronoun (*ze*). This phenomenon is much more frequent in historical (and dialectic) Dutch than in German. Clitics may be attached to other word classes like conjunctions and more than one clitic can be attached to a single word (cf "*indienmense*" ~ german *indem man sie*).

Clitic combinations will be lemmatized by assigning an ordered sequence of lemmata. A word form like '*sboexs*' (= des Buches) will thus be lemmatized (HET, BOEK). In the database, the ordering will be reflected by assigning a sequence number tot the lemma parts (see Figure 2 and Tabel 2). Each part will have its own part of speech. Thus, the complete Lemma-PoS assignment for sboexs will be

Sboexs~ {(1,*HET/DAT*, PRN), (2, *BOEK*, NOU(infl=s)}.

The sequence numbers are included to distinguish between words like *'kzag* and *zagk*.

Comment
This treatment of clitic combinations serves the following purposes: the lemma parts can be used as search keys, while the combination of all parts serves as a variation-independent key grouping different realizations of basically the same clitic combination.
A segmentation of the clitic combination as a sequence of word forms is not included in the database because this is, in many cases, problematic because of sandhi phenomena, cf. middle dutch *dat = dat + het*, Middle German *deist = da + ist.*
Clitic combinations are very common in Italian and Spanish (*damelo = da+me+lo*, give-me-it. Of course, they are quite common in Middle German (deist = da+ist, enloufen = en (not) + laufen, etc.).

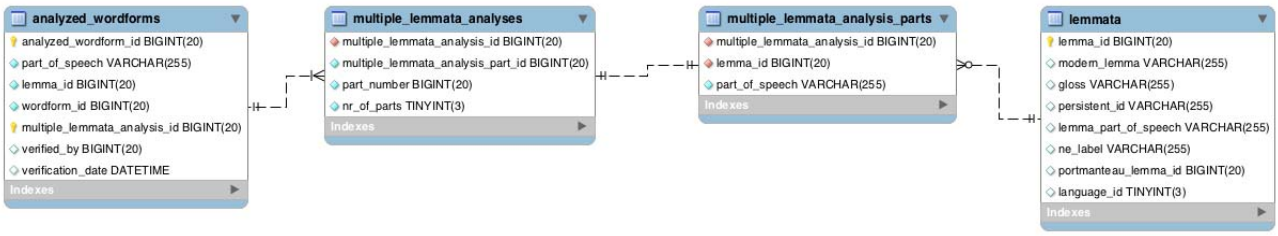


Figure 2. Multiple lemmata analysis.

Table 2: example data for an analyzed clitic combination

Table lemmata

Lemma_id	modern_lemma	lemma_part_of_speech
L1	Ik	PRN
L2	Zij	PRN
L3	Zien	VRB

Table analyzed_wordforms

Analyzed_wordform_id	Pos	part_number	Multiple_lemma_analysis_id	lemma_id	wordform_id
A1	CLITIC	NULL	Mla_1	NULL	W00001

Table multiple_lemmata_analyses

Multiple_lemmata_analysis_id	multiple_lemmata_analysis_part_id	Nr_of_parts	Part_number
Mla_1	Mlap_1	3	1
Mla_1	Mlap_2	3	2
Mla_1	Mlap_3	3	3

Table multiple_lemmata_analyses_parts

Multiple_lemmata_analysis_part_id	multiple_lemmata_analysis_part_id	Part_nr	POS	Lemma_id
Mlap_1	Mla_1	1	PRN	L1
Mlap_2	Mla_1	2	VRB	L3
Mlap_3	Mla_1	3	PRN	L2

Table wordforms

Wordform_id	Wordform
W1	‘ksachse

Status of this information: mandatory when applicable (when clitic combinations are prominent in the language, something has to be done about them). Use: internal and external (TR5 has to be able to deal with the clitic combinations as well).

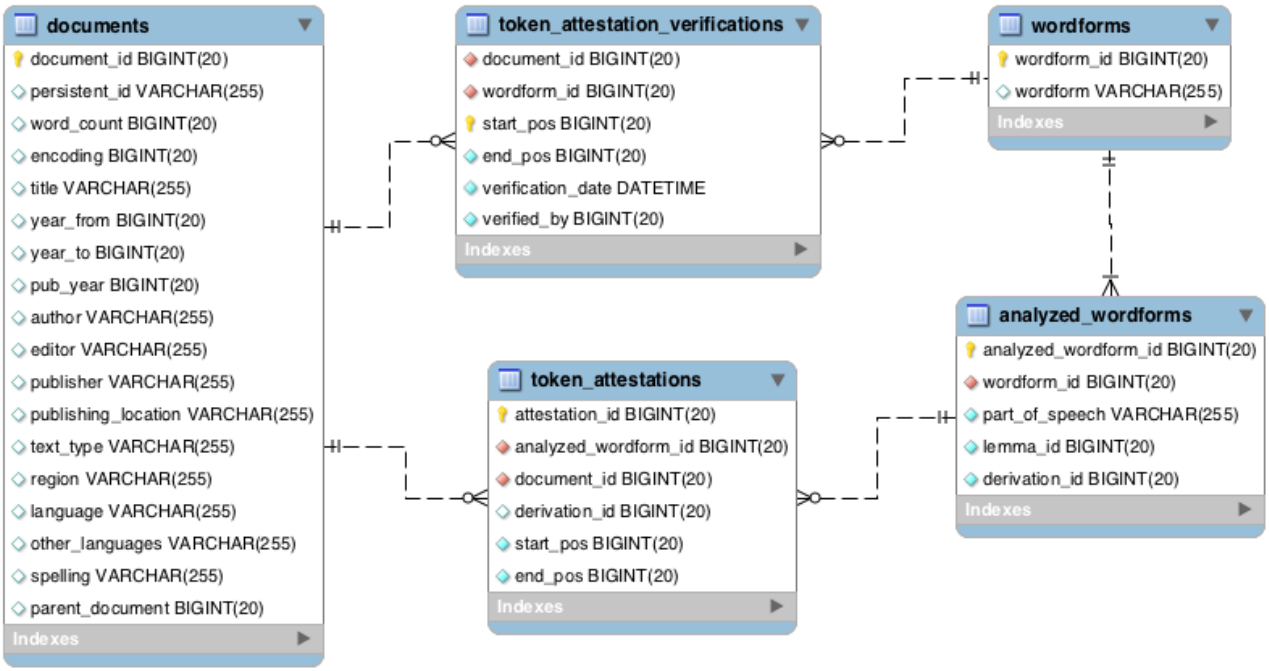
2.5. Attestation

One of the most important tasks for the IMPACT lexicon building process is to keep track of the origin of word forms. An unstructured, ever-growing set of word forms, without information about the kind of text (in terms of period and subject matter) in which we can expect the words to occur, is neither usable in text recognition nor in enrichment. Hence, to each labeled or unlabeled word form, we link *attestation* objects which are basically just verified occurrences of the words in documents. The attestations enable us to derive the relevant information about the domain of applicability of word forms from the properties of the documents they occur in.
When a word form is taken from a lexicon or dictionary, or it originates from automatic analysis expansion, we also keep track of its provenance. This is covered in the next section.

Besides the link to the relevant word form and a location in a document, the attestation objects contain the following information:

- Verification (yes/no): Is the occurrence of a labeled word form checked manually by an expert?
- Frequency in a document or document collection

Several distinct kinds of attestation may be relevant: we may just link a word form to a document, recording the frequency of occurrence ("attestation at text level"), or we may link to an individual occurrence of the word ("attestation at the token level")[8]. The latter kind of attestation is especially relevant to tagged corpora. In the lexicon building workflow, lemmata may first be assigned on the text level, and ambiguity is not completely resolved. At a later stage, ambiguity may be resolved by assigning lemmata on the token level.



[8] A type is a word form, a token is a particular instance (occurrence) of the type in a text.

Figure 2: database model for the attestation of word forms in documents[9]

2.5.1. Attestations on the token level

The representation of a lemmatized fragment in the database:

Everybody is loved by somebody?

Table lemmata

lemma_id	modern_lemma	lemma_part_of_speech
l1	EVERYBODY	PRN
l2	BE	VRB
l3	LOVE	VRB
l4	LOVED	ADJ
l5	BY	ADP
l6	SOMEBODY	PRN

Table wordforms

wordform_id	Wordform
wf1	Everybody
wf2	Is
wf3	Loved
wf4	By
wf5	Somebody

Table analyzed_wordforms

Analyzed_wordform_id	Part_of_speech	lemma_id	wordform_id
ana1	PRN	l1	wf1
ana2	VRB(3sg)	l2	wf2
ana3	VRB(part)	l3	wf3
ana4	ADJ	l4	wf3
ana5	ADP	l5	wf4
ana6	PRN	l6	wf5

Table token_attestations

Attestation_id	Quote	Analyzed_wordform_id	Document_id	onset	offset
1	NULL	ana1	text1	0	8
2	NULL	ana2	text1	9	11
3	NULL	ana3	text1	12	17
4	NULL	ana4	text1	12	17
5	NULL	ana5	text1	18	20
6	NULL	ana6	text1	21	29

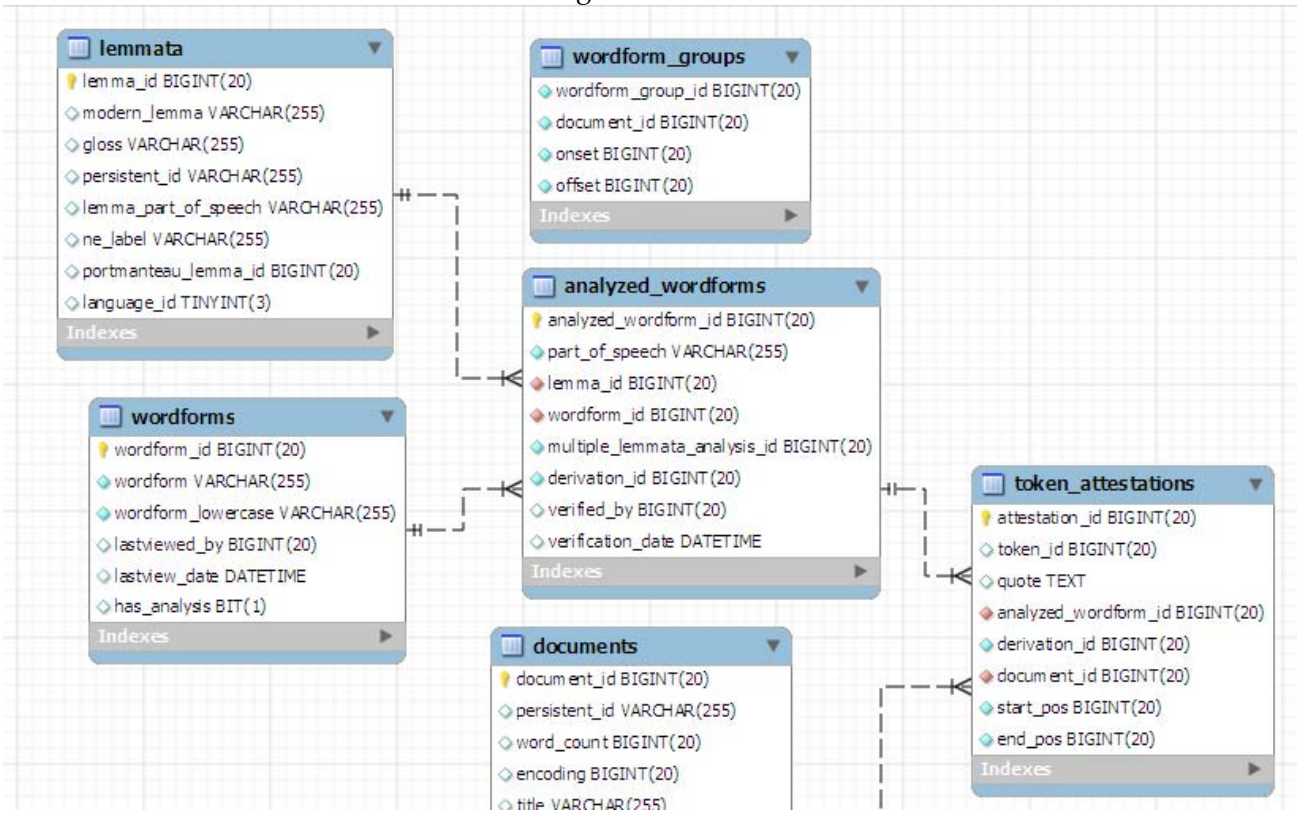
2.5.1.1. Token group attestations

There are several ways in which a group of "graphical" tokens can be linked to a single analysis of the

[9] We only give the diagram for attestations of labeled word forms. The diagram for attestations of unlabeled word forms is completely analogous.

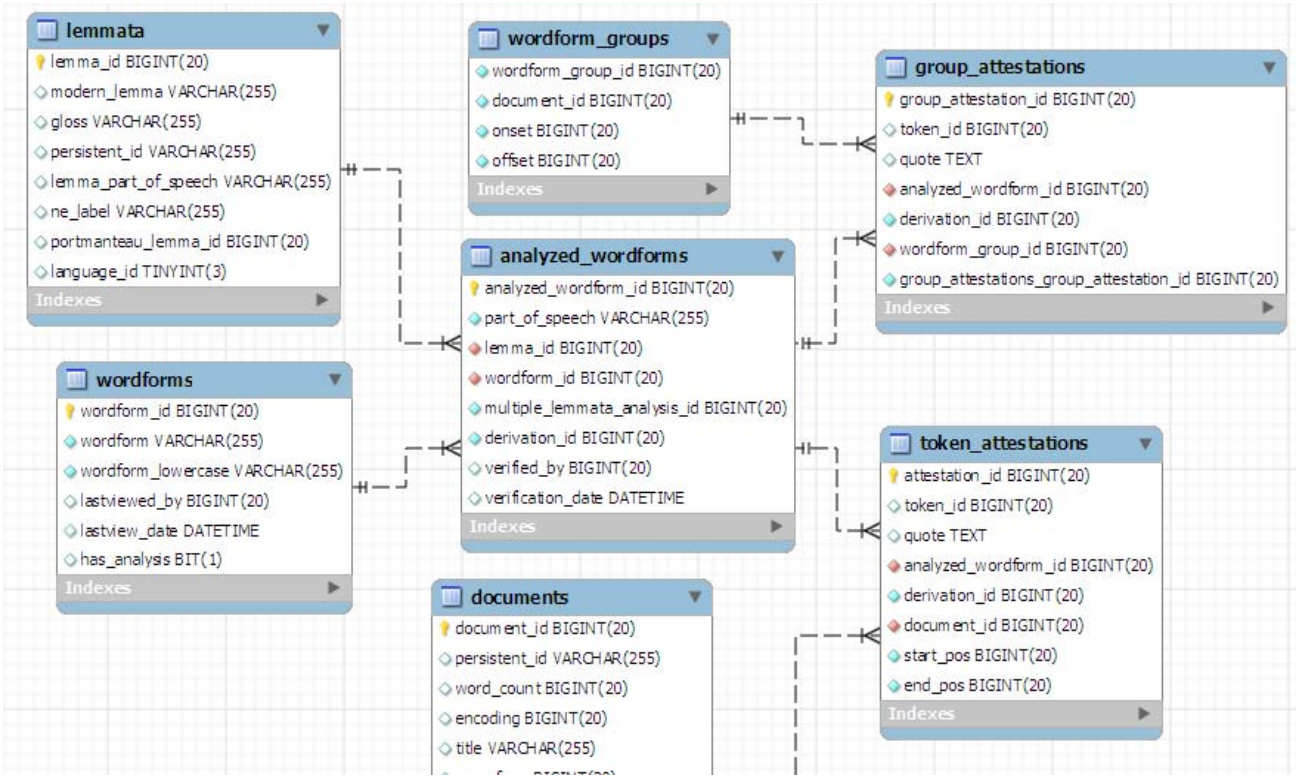
group as a whole.

1. Two or more tokens are joined by a "wordform group"; the group is analyzed as a whole; This is typically the case for more or less "accidental" split realizations of non-compound word forms like "gelopen" as "ge lopen";
   In these cases all group members get the same analysis as a token attestation and all group members are mentioned in the wordform_groups table. The group_attestations (cf. 2. below) table is not used in this case. The following tables are used:



2. The tokens are joined in a wordform group; but the individual tokens have an analysis of their own. This is applicable to
   1) Idiomatic expressions (not tackled as such in IMPACT)
   2) Multiword named entities. E.g. *Benedykta Chmielowskiego* is analyzed as a compound word form for the NE lemma *Benedykt Chmielowski*; but we also do not wish to omit the information that *Benedykta* belongs to the lemma *Benedykt*, and *Chmielowskiego* belongs to *Chmielowski*.

The following structure is present in the database for this purpose: *wordform_groups* serves to link several tokens by a single group id. *Group_attestations* gives the possibility to link such a group of tokens as attestation data to *analyzed_wordforms.*



To give an example, suppose we have the following short sentence:

To Jest Przez Xiędza Benedykta Chmielowskiego Dziekana Rohatyńskiego, Firlejowskiego, Podkamienieckiego Pasterza.

Token	"raw" token with punctuation	Character offset of start of token	Character offset of end of token
To	To	0	2
Jest	Jest	3	7
Przez	Przez	8	13
Xiħdza	Xiħdza	14	20
Benedykta	Benedykta	21	30
Chmielowskiego	Chmielowskiego	31	45
Dziekana	Dziekana	46	54
Rohatyqskiego	Rohatyqskiego,	55	68

lemmata

lemma_id	modern_lemma	lemma_pos
1	to	PRN
2	byf	VRB
3	przez	ADP
4	Ksidz	NOU
5	Benedykt	NOU
6	Chmielowski	NOU
7	Benedykt Chmielowski	NOU

wordforms

wordform_id	Wordform
wf1	To
wf2	Jest
wf3	Przez
wf4	Xiħdza
wf5	Benedykta
Wf6	Chmielowskiego
Wf7	Benedykta Chmielowskiego

analyzed_wordforms

Analyzed_word_form_id	Pos	Multiple_lemma_analysis_id	lemma_id	wordform_id
A1	PRN	NULL	1	Wf1
A2	VRB	NULL	2	Wf2
A3	ADP	NULL	3	Wf3
A4	NOU	NULL	4	Wf4
A5	NOU	NULL	5	Wf5
A6	NOU	NULL	6	Wf6
A7	NOU	NULL	7	Wf7

wordform_groups

Wordform_group_id	document_id	onset	offset
1	text1	21	30
1	text1	31	45

group_attestations

Group_attestation_id	analyzed_wordform_id	Wordform_group_id
1	Ana7	1

token_attestations

attestation_id	analyzed_wordform_id	document_id	start_pos	end_pos
1	A1	text1	0	2
2	A2	text1	3	7
3	A3	text1	8	13
4	A4	text1	14	20
5	A5	text1	21	30
6	A6	text1	31	45

2.5.2. Attestations on the text level

This type of attestation is linked to the occurrences of a word in text, without specifying the location in the document. It is important in our workflow that also partially disambiguated information can be stored and used. i.e. several attestations may be linked to the same type or token.

Table text level attestations.

Attestation_id	Frequency	Verified	Analyzed_wordform_id	Document_id
Tla1	23	True	A100	Text1

2.5.3. Verifying non-analyzed word forms

In some context word forms can be attested for which no analysis is available. For this reason the table 'token_attestation_verifications' is introduced in the database. Attestations of this type link directly to wordforms.

In some cases the annotator might decide not to assign a lemma to the token. The token is then marked as verified. Verified tokens might be revisited at a later stage.

Status of attestation information: mandatory, external (for use in TR5), and for internal use in EE2 and EE3

2.6. Derivations

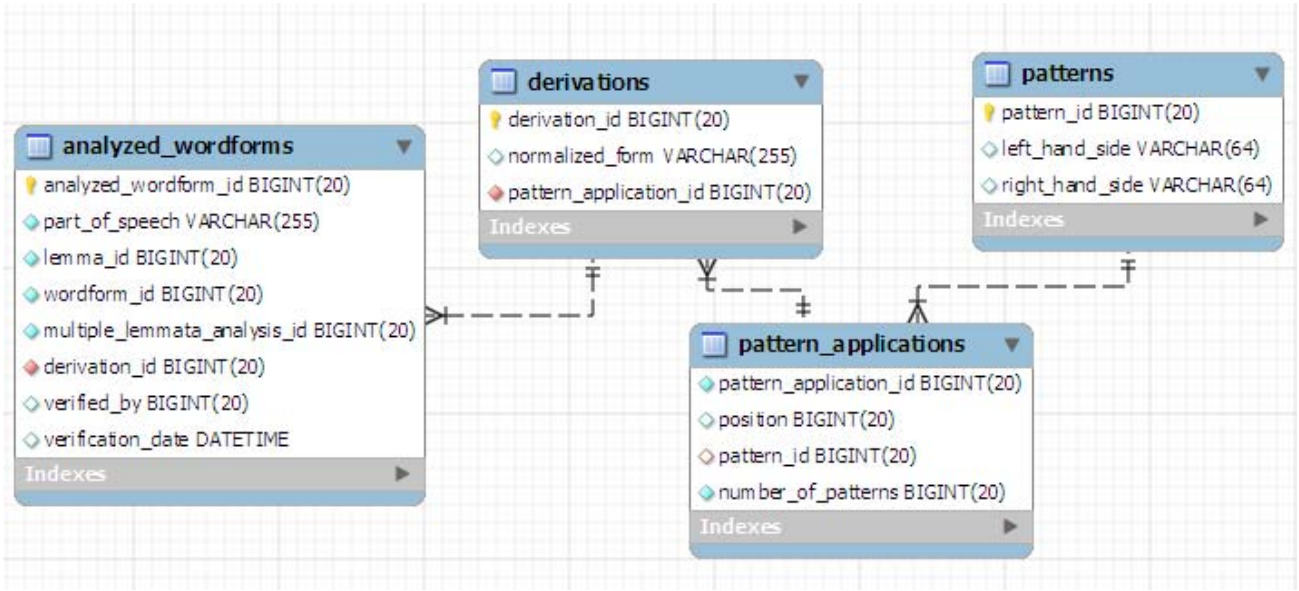


Figure 3. Derivations

Word forms can get a more elaborate analysis than just a part of speech and a gloss. A modern word form can be attached, and possibly also a set of patterns that describes how to get to the older word form from the modern one. E.g:

theyle, <teile>, [(t_th,0), (ei_ey, 1)], NOUN, teil

Here the part between angled brackets (<>) describes the modern word form, and the part between

square brackets ([]) describes the patterns.

Table derivations

derivation id	Ìdentifier of the derivation
normalized_form	The modern word form. Can be NULL.
pattern_application_id	Identifier of pattern application if applicable. Can be empty, in which case it is 0 (nil, not NULL)

Table pattern_applications

pattern_application_id	Identifier of the derivation. NOTE that this is NOT a primary key. Rather it is used to group sevaral patterns together. The unique key of this table is composed of all the field together.
Position	The position in the string that the pattern is applied to (0 and 1 in the example)
number_of_patterns	The amount of patterns that go with this analysis (two in the example above). This number is in a way redundant, because it is always the same as the amount of records sharing the same identifier. Storing the number here however makes some queries a lot faster and easier.
pattern_id	Identifier of the pattern associated.

Table patterns

pattern_id	Ìdentifier of the pattern
left_hand_side	The left hand side of the patterns. What is left of the underscore. So 't' and 'ei' in the example above.
right_hand_side	The right hand side of the patterns. What is right of the underscore. So 'th' and 'ey' in the example above.

Please note that both patterns and modern word forms can be empty.
In other words

theyle, [(t_th,0), (ei_ey, 1)], NOUN, teil

theyle, <teile>, NOUN, teil

are both valid analyses.

If there are patterns but no modern word form (as in the first example above), a row in the derivations table is created none the less to tie the patterns and the analyzed word form together. Its modern word form field will be left empty however.
If there is no pattern but a modern word form is provided (as in the second example above) then there will just be a row in the derivations table and no corresponding pattern applications nor patterns.

2.7. Documents, corpora and workflow management

In order to record provenance details, the database is provided with the structure depicted in Figure 4.

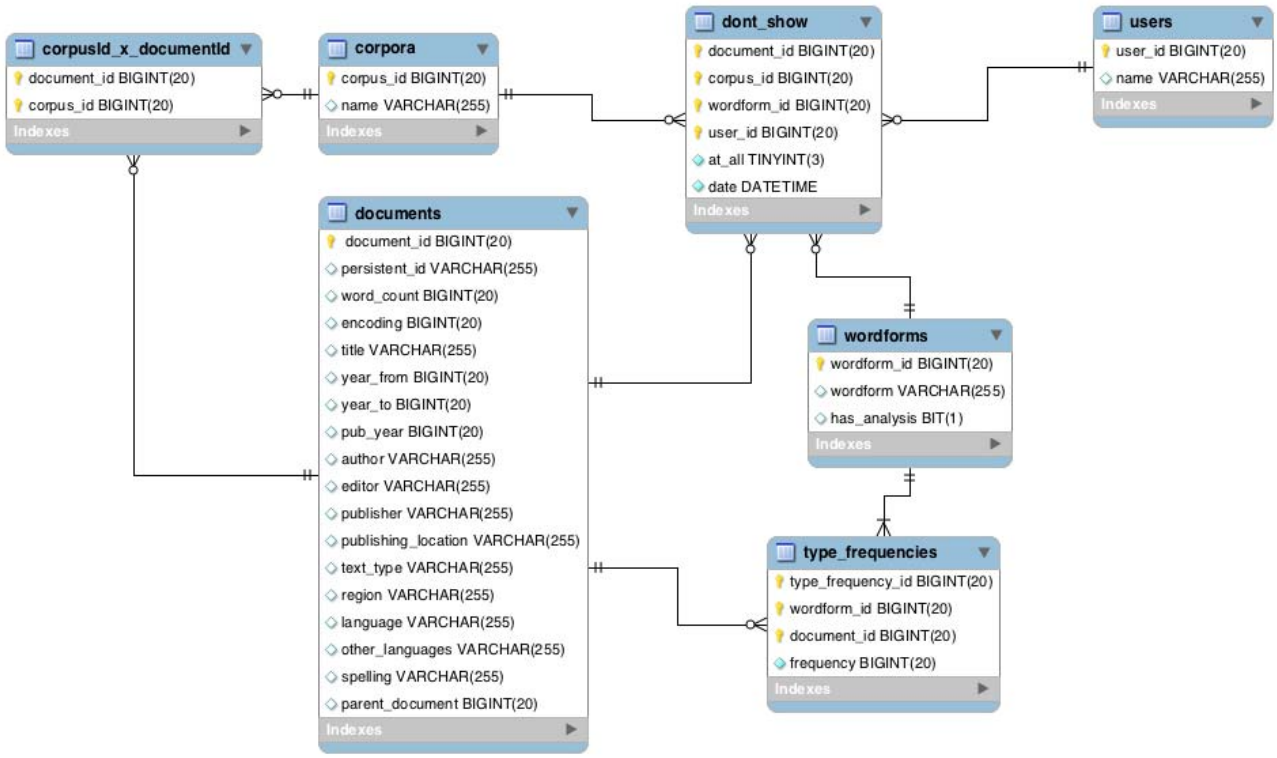


Figure 4. Documents, corpora and workflow tables.

Documents can be organized in corpora. An important reason for this is the allocation of properties to a large number of documents at once.
The table 'type_frequencies' contains the relations between word forms and documents. When a document is to be annotated, all of its word forms are added to the table 'wordforms' (unless there already exists an entry for that word form). Simultaneously, the frequency of the word forms occurring in the document is registered in table 'type_frequencies'.
The table 'dont_show', can be used during the building of the lexicon. Certain word forms (e.g. frequent function words) should not be presented to the annotators over and over again during the process of attesting documents and corpora. It is possible to exclude certain wordforms from attesting in a certain document, in a certain corpus, or in all documents and corpora.

Table dont_show

Wordform_id	Document_id	Corpus_id	At_all	User_id	Date
Wf201		SG1873		1	15-01-2010

For administrative purposes we added a table 'users'. Here we register staff members who are tasked with manual annotation and verification.

Table users

User_id	name
1	Jan van der Wiel

3.Information attached to lemmata

Lemmata are linked to word forms (cf. 2.4). In their turn, lemmata need several other information

categories to fulfill their role in the lexicon, which will be described in this section.

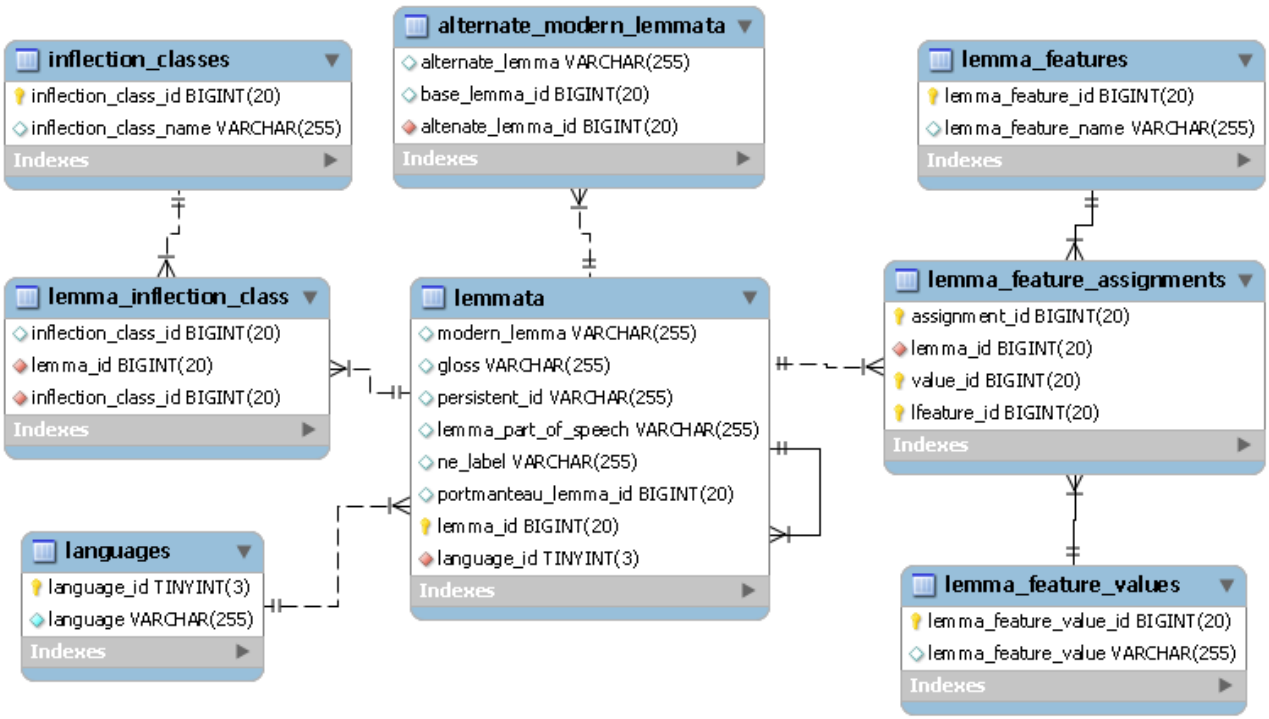


Figure 5: basic lemma information

3.1. Lemma-id

It goes without saying that each lemma is assigned a unique ID.

Status: mandatory

3.2. Modern lemma form

Recall that the modern lemma form is used as a variation-independent search key in Enrichment. The general rule is to assign a single modern lemma form. In some cases, it will be profitable to add more than one modern lemma form, because several variants survive in the modern language, with more or less equal status. A separate table stores these variants. Typical examples in Dutch: *Weer/weder*, *neer/neder*.
There will be a separate document about the principles of assigning a modern lemma to historical word forms.

Status of this information: mandatory, both for internal and external use. Modern lemma variants are optional.

3.3. Lexical part of speech

A main part of speech is assigned to each lemma (e.g. NOUN, VERB, ADPOSITION, ….).
Part of speech is not by itself a deliverable of IMPACT, but the lexicon cannot be organized without it.
Part of speech distinguishes lemmata. Additional features (like gender, inflectional class) do not by themselves constitute a sufficient criterion to distinguish lemmata, since they are very much subject to historical variation (e.g.: at least 3815 nouns from the Dutch Woordenboek der Nederlandsche Taal have more than one possible gender). We do not specify which additional features may be used for all different languages. Instead, we provide a general mechanism for adding features (cf. 3.6 and 3.4).

Status of this information: mainly for internal use[10], but hardly dispensable as a means to organize the lexicon, so: mandatory.

3.4. Gender and other possible grammatical features

Gender information is important as an organizational principle in, for instance, German. In other languages, features like animate/non-animate may be relevant.
In languages with poor inflection morphology, it is often possible to have several genders for a single lemma. Hence the suggested general feature assignment mechanism (cf. figure 1).

Example: gender

Table lemma_features

lemma_feature_id	Lemma_feature_name
1	Gender
2	Foreign_Language_Name

Table lemma_feature_values

lemma_feature_value_id	lemma_feature_value
1	M
2	V
3	French
4	German

Table Lemma_feature_assignments

assignment_id	feature_id	value_id	lemma_id
1	1	1	19289
2	1	2	19289
3	2	4	20001

Status: optional, internal, depending on the importance of these notions in the language at hand.
Remark: Within the NE context, this can be used to tag words as belonging to a foreign language (Koroška [SLOVENIAN]).

3.5. Named entity label

For named entities (NE), either multiword or single, a classification label is added according to the

[10] Part of speech tagging is not a deliverable of IMPACT

scheme chosen for IMPACT. The proposed labels are NE-PER (persons), NE-LOC (locations), NE-ORG (organizations).

Status of this information: for internal and external use, mandatory.

3.6. Inflectional class(es)

Inflectional classes are necessary for the basic generation of word forms in the reverse lemmatization task.

Status of this information: for internal use, but hardly dispensable as a means of organization.

3.7. Language

When a text contains words from another language, they should be marked accordingly.

3.8. Gloss

Lemmata may have a short description of word meaning. This is especially relevant to be able to distinguish between homographs.

Status: optional, internal and external use.

3.9. Multiword expressions

The inclusion of multiword expressions (MWE) takes us to the boundary of syntax and morphosyntax. A lot of recent research has been devoted to the position of multiword expressions in the lexicon; much of this work is concerned with the syntactic treatment or the semantic interpretation of idioms, which is decidedly out of scope for IMPACT.
Within IMPACT, MWE are likely to play a role for named entities and for constructions which can be realized both as a single orthographic token and as several tokens (e.g. separable verbs and detached word parts).

There are two distinct ways of adding multiword structure to the database. We can map a multiword expression realized as a word form to a sequence of lemmata and PoS labels by using the structure already present in the database for the storage of clitic combinations (cf. 2.4), and the constituent parts of multiword lemmata are specified using a mechanism parallel to the way we treat morphological analysis (3.10). Some typical cases:

1. Transparent: there is a clear 1-1 correspondence between the parts of the word form, separated by whitespace, and the lemma parts. *Karl der Grosse*, *Karls des Grossen*.
   Most naturally seen as a sequence of word forms, each with their own lemma and PoS. The sequence has a higher-level PoS and lemma as well. Cf. also 2.5.1.1.
2. Non-transparent: 'zu ruck': two typographic words but just one 'linguistic' word form (containing whitespace, no special treatment required in the lexical database).
3. Some combinations like Middle Dutch '*al die wile dat*' (*all the while that*): admit for both points of

view. The fact that the combination occurs with different typographical segmentations (cf. examples below) points to an analysis along the lines of the analysis of clitic combinations. (Dictionary of Early Middle Dutch:

Ende ***al de wile dat*** soe drinct yet Sone drinc en twint selue niet, *En.Cod. p. 486-487, r. 42-6, Oost-Vlaanderen, 1290*

Ende ***aldie wile dat*** si ghingen olie koepen, so quam die brudegoem, ende die gheret waren, ghingen met hem in ter brulocht, *Diat. p. 222, r. 12-16, Brabant-West, 1291-1300)*

The equivalence class method (ECM , Odijk 2004) is quite similar to what we intend to do on the lemma level. In order to arrive at a representation which can be used in different possible grammatical theories, Odijk proposes to include the following information for each idiom:

1. Idiom pattern id (=our multiword_operation_id)
2. Idiom component list (=multiword_analysis)
3. Example sentence (we should get this from the attestations)

In order to deal with inflected forms of multiword expressions, pattern equivalence can be defined such that equivalent multiword expressions have similar inflectional properties.

Status of multiword data: optional for the general lexicon; indispensable for the named entities lexicon.

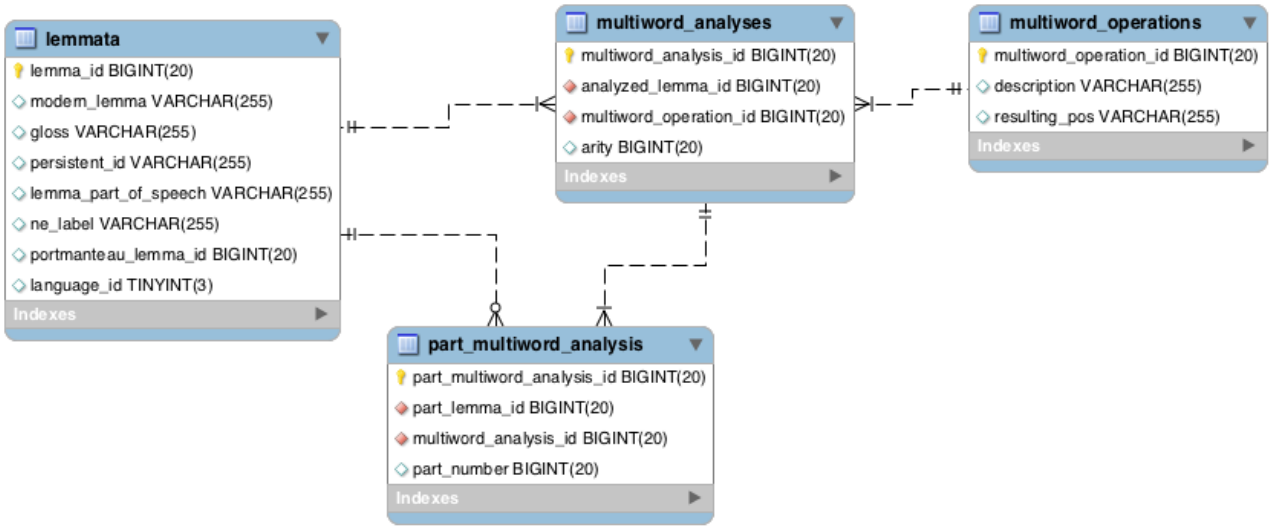


Figure 6: database model for multiword lemmata[11]

Table lemmata

lemma_id	modern_lemma	lemma_pos
L102	al-de-wijl-dat	CONJ
L501	Al	PRN
L502	De/die[12]	PRN
L503	Wijl	NOU

[11] For an explanation of the self-reference in the definition of the lemmata table, cf section 3.11.1, portmanteau lemmata

[12] The slash indicates alternatives

L504	Dat	CONJ

Table multiword_analyses

Multiword_analysis_id	Arity	Analyzed_lemma_id	multiword_operation_id
a102	4	L02	M1

Table part_multiword_analysis

Part_multiword_analysis_id	Part_number	Part_lemma_id	Multiword_analysis_id
P1	1	L501	a102
P2	2	L502	a102
P3	3	L503	a102
P4	4	L504	a102

3.9.1. Multiword named entity lemmata

The inclusion of Named Entities (NEs) in the lexicon is crucial in the sense that, on the one hand, text recognition is based on input from the lexicon, so we want to capture as many possibly occurring tokens as possible, and on the other hand, names of persons, organizations and places are very likely candidates for users' search queries, hence, normalizing them with respect to orthographical and interlingual variation is desirable.

NE's can occur in the form of multiword expressions or as single tokens. In principle, the mapping from multiword NEs to lemmas works the same way as with idiom parts, i.e., the entire complex receives a Lemma ID, and the parts are mapped onto their corresponding lemma's, if available. For the possible values of the property "NE label" cf. section 3.5. For the treatment of wordforms and attestations for multiword NE's, cf. 2.5.1.1.

Table lemmata

lemma_id	Modern_lemma	lemma_pos	ne_label
L202	Jan van de Wiel	NOU	NE_PER
L601	Jan	NOU	NE_PER
L602	Van	ADP	
L603	De	PRN	
L604	Wiel	NOU	NE_PER

Table part_multiword_analysis

Multiword_analysis_id	Arity	Analyzed_lemma_id	Multi_operation_id
A202	4	202	m1

Table part_multiword_analysis

Part_multiword_analysis_id	Part_number	Part_lemma_id	Multiword_analysis_id
P1	1	L601	A202
P2	2	L602	A202
P3	3	L603	A202
P4	4	L604	A202

Note that there is no distinction between single-word and multiword NE's, as both types are identified as the same PoS category, and that the person name *Wiel* is not mapped to the noun *wiel* ('wheel').

Status of NE information: mandatory, internal and external use

3.10. Morphological analysis

This section is about derivation and composition. The paradigmatic relation between lemma and word forms is treated in section 2.4. Morphological analysis will be attached at the lemma level. The word forms belonging to lemma's will inherit this analytical information.
Within IMPACT, morphological analysis is not a purpose of its own, but serves practical ends:
- to function in a spellchecker that does not reject newly found productive compounds because of their deviant forms
- analysis of existing compounds can be used to predict inflectional forms for compounds/ word forms which will be generated automatically (expansion).

Some remarks:
1) Morphological analysis can be specified in the form of a full hierarchical analysis, or a flat list of components, or (partial analysis) one can just specify the head of the compound, which usually determines its morphosyntactic properties. The proposed database structure is compatible with these three possibilities. We want to stress the idea that different solutions are possible for different languages.
To fulfill practical ends, we don't always need full blown deep analyses. We only have to be able to say which type of compound and which final parts of a compound are very frequent.

- A 'deep' analysis can be obtained by storing, recursively, the analyses of the immediate constituents (*Braumeisterfleischpflanze* is analyzed as a nominal compound of *Braumeister* + *Fleischpflanze*, a deeper analysis can be stored if *Braumeister* is analyzed in its turn as *brauen+Meister* and *Fleischpflanze* as *Fleisch+Pflanze*, etc.).
- An (arbitrarily long) "flat" analysis of a compound is also possible, Braumeister+Fleisch+Pflanze. There is often no need to choose between different 'bracketings' of a compound.
- If the focus is on predicting the morphosyntactic properties of the compound, it is sufficient to analyze this word as "nominal compound with last part *Pflanze*". It is NOT mandatory to link to all parts of a compound.

2) Diminutives are assigned lemmata of their own but the relation to the base lemma is stored.
3) It is allowed to assign more than one analysis of one compound lemma.

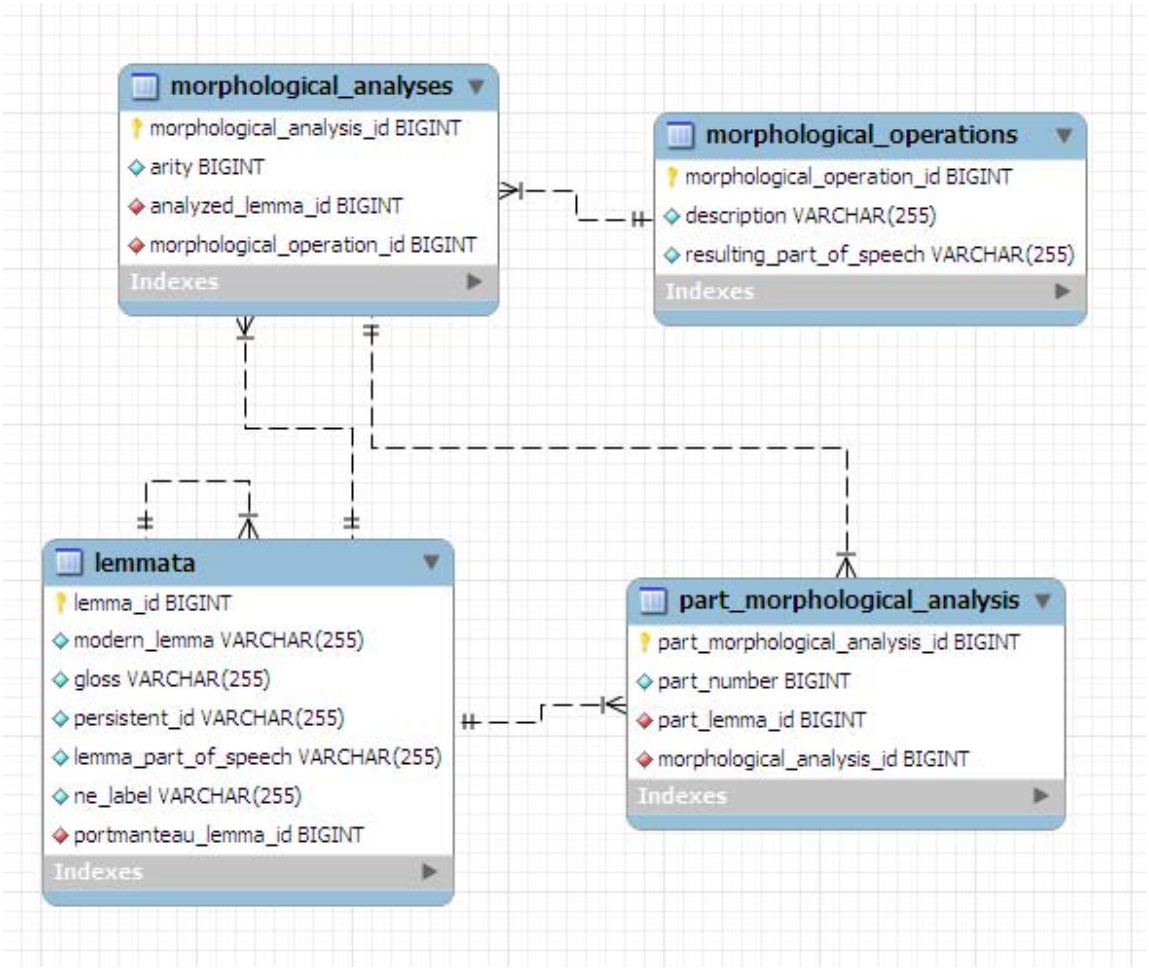


Figure 7: database model for morphological analysis

Status of morphological analysis: internal + external, optional (in the sense that not all words must be analyzed)
Internal use: use to predict paradigms of compounds and derivations
External use: use in OCR to help assess the probability of unknown words

Table 1: database examples for morphological analysis

Table lemmata

Id	Modern_lemma	Lemma_pos
L000001	Appelflap	NOU
L000002	Appel	NOU
L000003	Flap	NOU
D1	Braumeisterfleischpflanze	NOU
D2	Braumeister	NOU
D3	Pflanze	NOU
D4	Brauen	VRB

D5	Meister	NOU
D6	Fleisch	NOU
D7	Fleischpflanze	NOU

Table morphological_analyses

Morphological_analysis_id	Arity	Analyzed_lemma_id	Morphological_operation_id
A1	2	l000001	o1
A2	2	d1	o1
A3	2	d2	o2
A4	2	d7	o1
A5	1	L000001	o3

Table morphological_operations

Morphological_operation_id	description	resulting_pos
O1	NOU+NOU->NOU	NOU
O2	VRB+NOU -> NOU	NOU
O3	.* + NOU -> NOU	NOU

Table part_morphological_analysis

Part_morphological_analysis_id	Part_number	Part_lemma_id	Morphological_analysis_id
P1	1	L0000002	A1
P2	2	L0000003	A1
P3	1	D2	A2
P4	2	D7	A2
P5	1	D4	A3
P6	2	D5	A3
P7	1	D6	A4
P8	2	D3	A4
P9	1	d3	A5

Note: Analyses a2, a3, a4 constitute a hierarchical analysis ((Brau)(meister))((fleisch)(pflanze)), a5 is a 'flat' analysis (brau_meister_fleisch_**pflanze**) which only links to the 'head' of the compound.

3.11. Unresolved ambiguity in lemma assignment

There are various ways of dealing with ambiguous word forms in the database. The basic mechanism is always the same: different analyses are attached to a single word form. This makes it possible to either leave it like that and not resolve the ambiguity at all or resolve it partially or resolve it completely. This depends on the requirements for the task.
The two mechanisms described below mainly serve to distinguish ambiguities which need not be resolved in IMPACT from other ambiguities which possibly do require a partial resolution.

3.11.1. Portmanteau lemmata

A portmanteau lemma is a lemma representing a group of homographs.
The purpose of portmanteau lemmata is to avoid choosing between two homographic lemmata (with equal modern lemma form and PoS), but different in meaning, inflection class or gender.
Portmanteau lemmata will be implemented as ordinary lemmata, linked to the homographs.
Cf. *heer*[1] (lord), *heer*[2] (army) or *bank*[1] (couch), *bank*[2] (bank), *Wetter*[1] (person who places bet), *Wetter*[2] (weather).
Portmanteau lemmata can be used to avoid complete disambiguation in morphological analysis as well: cf. *tuinbank* vs. *handelsbank* or *heerbaan* (*heirbaan*) vs. *heerendas*. A word form which belongs unambiguously to the paradigm of one of the homographs can be assigned directly to the more specific lemma, e.g. the old form *'har'* belongs only to *heer*[1].
NB: portmanteau lemmata will not be used to group homographic lemmata with different PoS. Cf. the discussion of 'transcategorization' (conversion), section 3.11.2.
Portmanteau lemmata were introduced for practical reasons:

- Lemmatizing a word form like Dutch *kip* to 15 possible homographic lemmata is not very attractive.
- How to update the ambiguous lemmatizations when another homograph is added to the lexical database?
- How to add data from a full form lexicon which need not have split the homographs in exactly the same way?

Status: optional, internal

3.11.2. Transcategorisation (conversion), sublemma and main lemma

Transcategorization (or *conversion*) occurs when part of the paradigm of a lemma X with PoS A can be seen as belonging to lemma Y with PoS B (e.g. participles, which can be seen belong to both a verbal and an adjectival paradigm). We call Y a sublemma corresponding to the main lemma X. In each language, we have a (small) fixed set of productive transcategorization relations. This list will be included in the database for the language.
While it might appear that including transcategorization information in the lexical database is linguistic hairsplitting and not relevant to IMPACT, it must be realized that it provides us with a principled way to avoid or defer decisions about lemma and PoS assignment to word forms like *geboren* (geboren/ADJ or gebären/VRB), or to leave the choice up to the user.

Figure 8: Database objects relating to morphosyntactic conversion (transcategorization):

Table 2: database examples

Table Lemmata

Lemma_id	Modern_lemma	Lemma_part_of_speech
L1	Bakken	VRB
L2	Gebakken	ADJ

Table Transcategorisations

Transcategorization_id	Mainlemma_id	Sublemma_id	Transcategorizationtype_id
T1	L1	L2	C1

Table Conversion_rules (List of transcategorizations present in the language)

Rule_id	main_pos	sub_pos	Transcategorisation_id
R1	VRB(part,past)	ADJ(infl=0)	C1
R2	VRB(part,past,infl=e)	ADJ(infl=e)	C1
R3	VRB(part,past,infl=en)	ADJ(infl=en)	C1

Table Transcategorisation_types

Transcategorizationtype_id	Description	main_pos	sub_pos
C1	Conversion between past participle and adjective	VRB	ADJ

Use of this data:

1) Lexicon expansion: create sublemmata automatically for non-incidental transcategorizations
2) Postponing or omitting disambiguation: distinguish between 'genuine' ambiguity (where for instance two semantically and etymologically completely different lexemes may be involved) and ambiguity resulting from different tagging principles

Status of this data: optional, internal

3.12. Adding custom information on the lemma level

If the database designer needs to store other lemma-related information, the recommended way is not to change the tables which are part of the basic structure, but to add tables linking the information to the relevant lemma ID's. If, for instance, it is desirable to add near-synonym information for retrieval purposes, the preferred solution is not to add fields to the *lemmata* table, but to add a table linking to it.

Example: *retrieval links* for near-synomyns or heads of compounds. Possible use: when searching for lemma_id, also search for related_lemma_id.

Table lemmata

Lemma_id	Modern_lemma	Lemma_pos
L001	Zange	NOU
L002	Seitenschneider	NOU
L003	Sonnenblume	NOU
L004	Blume	NOU

Table retrieval links

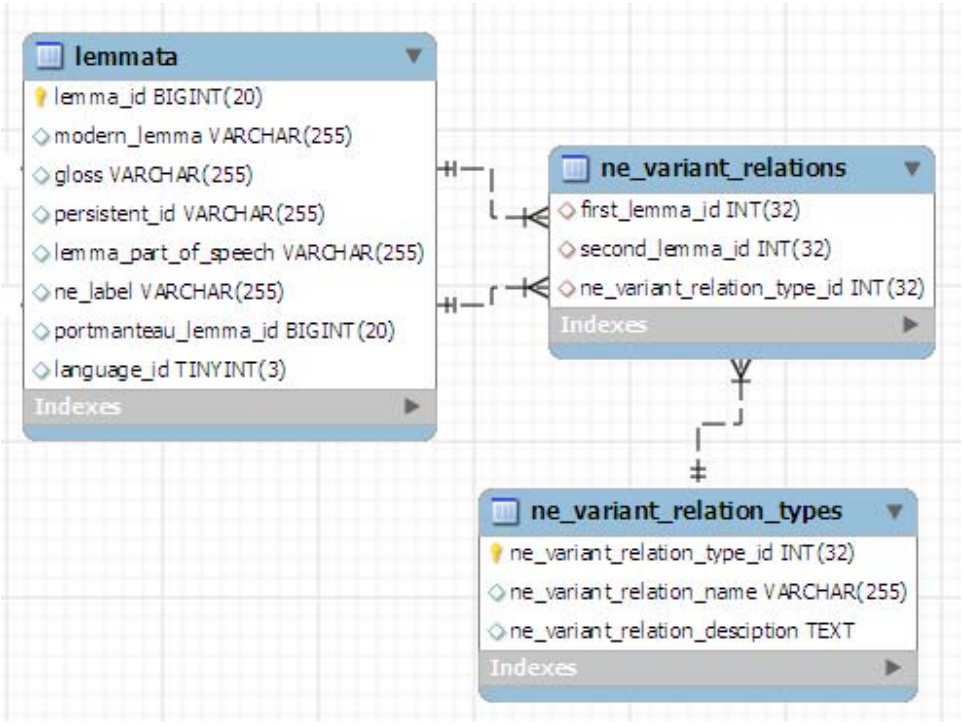
Lemma_id	Related_lemma_id
L001	L002
L004	L003

Status of this information: optional, mainly for external use in retrieval.

3.13. Additional structure for related entries in NE lexica

In the general lexicon, variants are included as word forms belonging to the modern standard lemma. This will also be the case for spelling variants of locations (Haerlem will be a word form with lemma Haarlem, etc).
For person names, however, we found it not feasible to distinguish between allographs of "the same name" and "etymologically related but different names". There are also variant relations like "interlingual variation" which deserve special treatment.

We propose the following structure:



Examples:

Table lemmata

Lemma_id	Modern_lemma	Lemma_pos	NE_Label
L001	Kärnten	NOU	NELOC
L002	Carinthia	NOU	NELOC
L003	Koroška	NOU	NELOC
L004	Douwes Dekker	NOU	NEPER
L005	Multatuli	NOU	NEPER

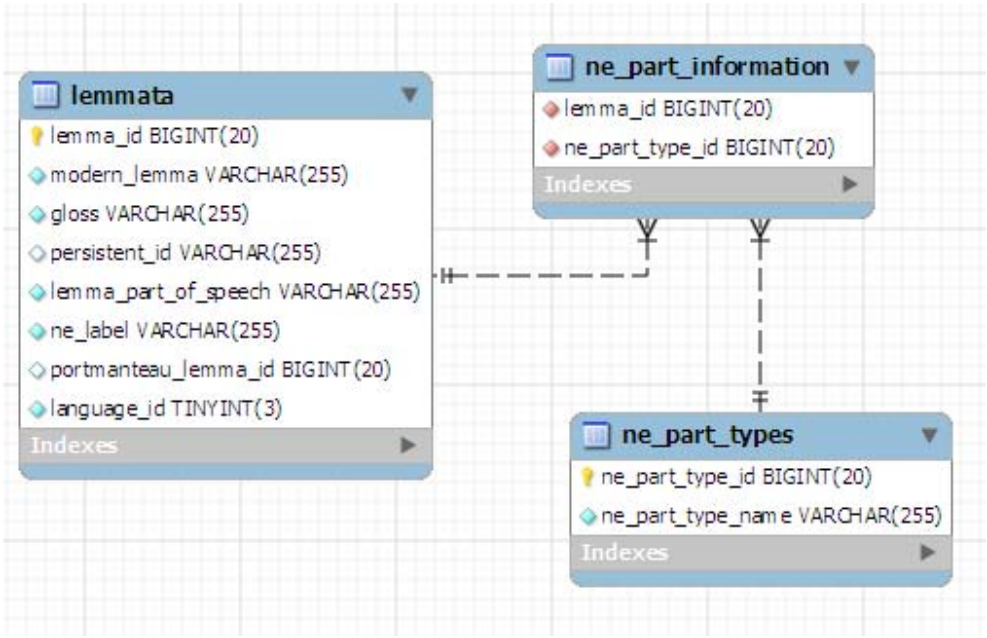
Table *ne_variant_relation_types*

Ne_variant_relation_type_id	Ne_variant_relation_name	ne_variant_relation_description
1	Interlingual_variant	Second is an other languafe variant of first
2	Pseudonym	Second is a pseudonym used by first

Table ne_variant_relations

first_lemma_id	second_lemma_id	ne_variant_relation_type_id
L001	L002	1
L001	L003	1
L004	L005	2

3.14. Named entity parts



These tables were added to allow parts of names tobe marked as such.
Examples of NE part types for Dutch are:

Givenname	Piet
Surname	Jansen
Title	dr., Jhr., baron
Particle	van, de, of, thoe, over, uyt
Suffix	junior, senior, sr. C.zn, A.zn, IIIe, Derde

Status of this information: optional, mainly for external use in retrieval.

4. Information on the document level

Information about the domain of application of words will be specified on the document level. By linking the words to the documents they occur in, they will inherit this information.
The following are relevant on the document level.

- Elementary bibliografical data:
  - Author
  - Editor
  - Title
  - Date of publication
  - Publisher
  - Publishing location
  - If document is part of e.g. a magazine, or a collected work, ...: reference to this work and to pages and/or issue/volume in this magazine, collection...
  - If document is in collection holder's catalogue: some ID or other type of link to the relevant item in the catalogue
- Text type, based on library metadata standards
- Number of words
- Date of text (can differ from date of publication, e.g. in case of editions
- Region of origin of text (dialect/language variety)
- character encoding (UTF8)
- primary language
- presence of other languages, e.g. Latin, French, ...
- To start with: informal description of the type of spelling used in the document. In the course of the project, this can be extended by a more formal profile. (f.i. Dutch: there is a difference between text material in the late nineteenth century spelling of *De Vries/Te Winkel* and the spelling of *Groene Boekje 1954.* Some authors e.g. Multatuli have their own spelling rules. This information is relevant.
- Location (path).

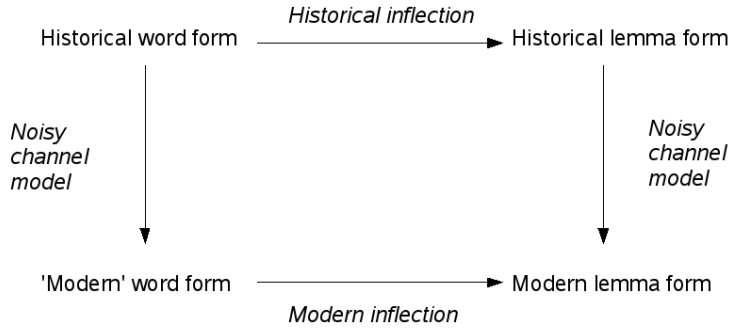
Status of this information: mandatory

5. Auxiliary information for word form synthesis and analysis

Another kind of information is the one for automatically generated or analyzed word forms,
Here, we keep track of:

- the inflection rule used
- the building element(s)
- the spelling patterns used to match the normalized spelling of the word form with the actual spelling in documents

The following diagram summarizes the relation between historical word form and modern lemma form, which is central in IMPACT lexica:



The horizontal axes correspond to models for inflectional morphology; the vertical axes correspond to spelling variation as it will be modelled in IMPACT[13].

5.1. Data to support the modelling of orthographic variation

In order to be able to induce statistical models for historical spelling by machine learning algorithms, some extra data, besides the relation of historical word form and modern lemma, must be developed. Without the 'modern word form equivalents', it is difficult to separate inflection from orthographical variation. The addition of this information is not entirely unproblematic. When there are morphological (and phonological) differences, a 'historical word form in modern spelling' may be a somewhat artificial construct. In manual annotation of ground truth material, we expect it to be much easier to choose a relevant lemma from a suggestion list of possible lemma assignments, than to choose a plausible transcription for an historical word form. There are, however, many cases where the differences between modern language and historical language are largely orthographic, and it is indeed possible to have some standard representation of historical word forms in modern spelling.

[13] The "noisy channel" models used assign weights to multi-character substitutions, thus defining a probabilistic model of orthographic variation.

The modern word form is useful, because a database of modern and historical word forms makes it easy to induce a set of patterns relating historical and modern spelling by a machine learning algorithm.
SUMMARIZING: It is of course not a problem to include this field in the database without being obliged to manually verify its contents. It may be sufficient to fill this in for only a relatively small number of word forms in a certain orthography in order to obtain the set of patterns needed to describe this particular orthography.

Example: the manual verification of the lemma assignment 'zeggen' to the historical word form seg(h)e is easy, choosing a modern form (zeg or zegge or even zeggen) is much less straightforward.

Position in paradigm	Middle dutch	Modern
1e sg.ind.pres.	sech, seg(h), segg, secge, seche, seg(h)e, segg(h)e	Zeg
1e pl.ind.pres.	secg(h)en, segg(h)en, zegghen, *directly followed by the pers.pron.* wi final-n often missing: secghe, segg(h)e, zegghe	Zeggen
imp.sg.	sech, seg(h), seg(h)e	Zeg
imp.pl.	sagit *(2x, Nederrijn)*, secget, sec(h)t, segg(h)et	Zegt
1e sg.conj.pres.	segg, segh, sage, segge *(it's not always certain that a conjunctive is involved)*	(nonexistent)
3e sg.conj.pres.	segg, sage, secghe, segghe	Zegge

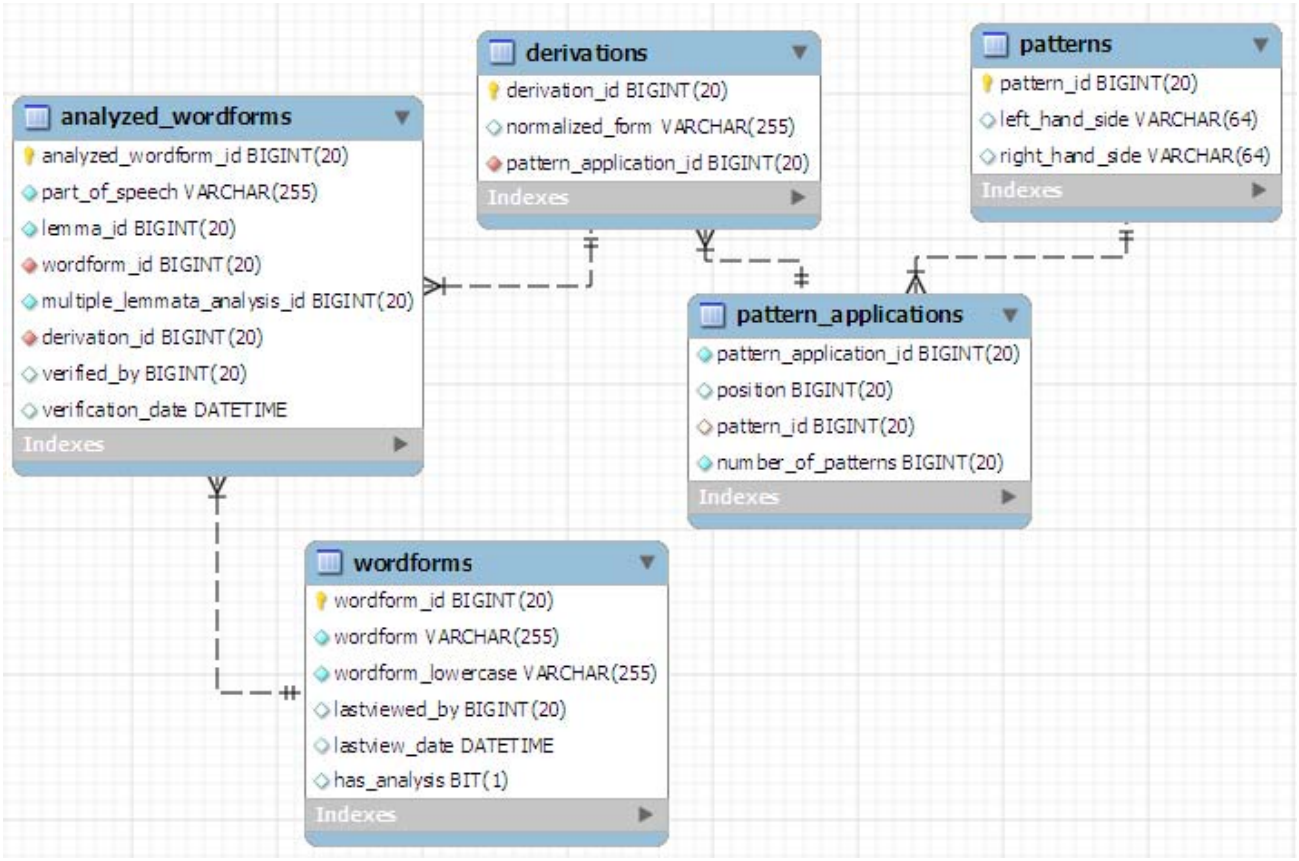


Figure 9. Derivations

Wordforms

Wordform_id	Wordform
W1	Klaerlick

Analyzed_wordforms

analyzed_wordform_id	Number_of_parts	Wordform_id
A1	0	W1

Derivations

Derivation_id	Normalized_form	analyzed_form_id
D1	Klaarlijk	A1

Patterns

Pattern_id	Left_hand_side	Right_hand_side
P1	aa	Ae
P2	Ij	I
P3	K	ck

Pattern_applications

Pattern_application_id	Position	Pattern_id	Derivation_id
Pa1	2	P1	D1
Pa2	6	P2	D1
Pa3	8	P3	D1

Status of this information: mandatory, external (for use within TR5)
The mandatory status of this information does not imply that it is completely manually verified. One may choose to generate this information from other information. The normalized word form may be chosen "on the fly"among the modern word forms of the lemma. The mapping, on the other hand, between historical word form and modern lemma is part of the deliverable output of the lexicon building process and the quality of this mapping has to be checked, and if necessary, manual corrections must take place.

5.2. Information about paradigmatic expansion

This is one way of keeping track of the process of expansion from lemmata to wordforms.

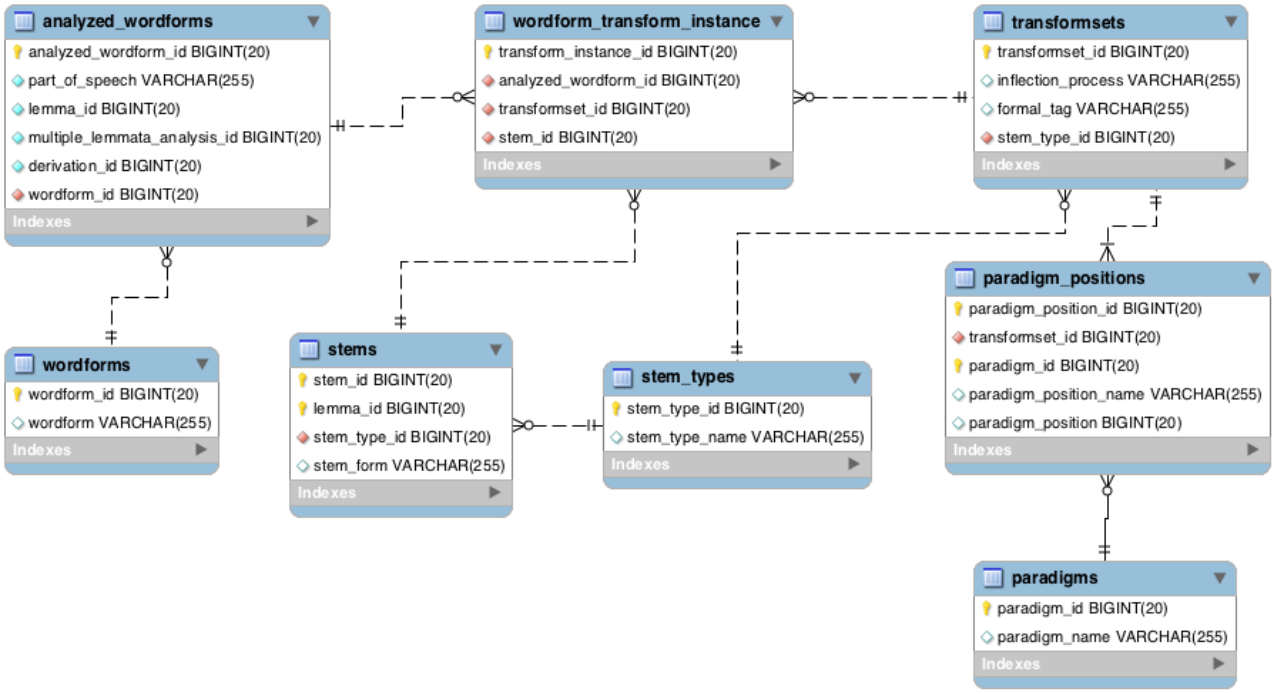


Figure 10. Paradigmatic expansion

Table paradigms

Paradigm_id	Paradigm_name
P1	Regular verbal a-stems
P2	Regular verbal e-stems

Table paradigm_positions

paradigm_position_id	paradigm_position_name	paradigm_position	Paradigm_id	Transformset_id
1	1 sg ind pres active	1	P1	R1
2	2 sg ind pres active	2	P1	R2

Table transformsets

Transformset_id	Inflection_process	Paradigm_position_name	Stem_type_id
R1	s/are$/o/[14]	1sg pres active a-stems	ST1
R2	s/are$/as/	1sg pres active a-stems	ST1

Comment: patterns for inflection may be either simple substitution rules or full-fledged finite-state transducers

Table wordform_transform_instances (defines the relation between inflectional patterns and word form instances)

Transform_instance_id	Transformset_id	Stem_id	Analyzed_wordform_id
R1	R2	Amare	A1

Table Wordforms

Wordform_id	Wordform
W1	Amas

Table analyzed_wordforms

analyzed_wordform_id	pos	part_number	number_of_parts	parent_analysis_id	lemma_id	wordform_id
A1	VRB(pres,2,sg,ind,act)	NULL	NULL	NULL	L1	1

Table Analyzed_wordforms

analyzed_wordform_id	Wordform_id	Number_of_parts
A1	W1	1

Table stems

Stem_id	Stem_form	Lemma_id	Stem_type_id
S1	Amare	L1	ST1

Table Stem_types

Stem_type_id	Stem_type_name
ST1	Lemma form

Status of this information: optional, internal

5.3. Database information for "stems"

It may not be very practical to derive the complete paradigm from a single base form (e.g. for strong or irregular verbs).
For this reason, we add a possibility to specify a number of alternate stem forms for a given lemma.

Table lemmata

Lemma_id	Modern_lemma	Lemma_pos
l1	Binden	VRB

[14] This example uses Perl 5 regular expression syntax

Table stems

Stem_id	Stem_form	Lemma_id	Stem_type_id
S1	Bind	L1	St1
S2	Band	L1	St2
S3	Bund	L1	St3

Table stem_types

Stem_type_id	Name
ST1	Present tense stem
ST2	Past tense stem
ST3	Past participle stem

Status of this information: optional, internal

6. Lexical source

For the existence of other words, no verified evidence in texts may have been found. It is still desirable to keep track of where they come from: incorporated from some other lexicon, obtained by expansion from lemmata in historical dictionaries, obtained by automatic (and not manually verified) analysis of historical documents. In the case of named entities, the lexical source information may serve to preserve the link to the persistent identifier in the library named authority data.
When information is incorporated from lexica or dictionaries, labeling from these sources may be copied (mapped – often a nontrivial task; subject matter labels may be useful; regional or temporal labeling may also be present) . Of course not all words in the source lexicon have identical date, text type, etc.. Hence in this case, the information is specified in the source information record for the word form.

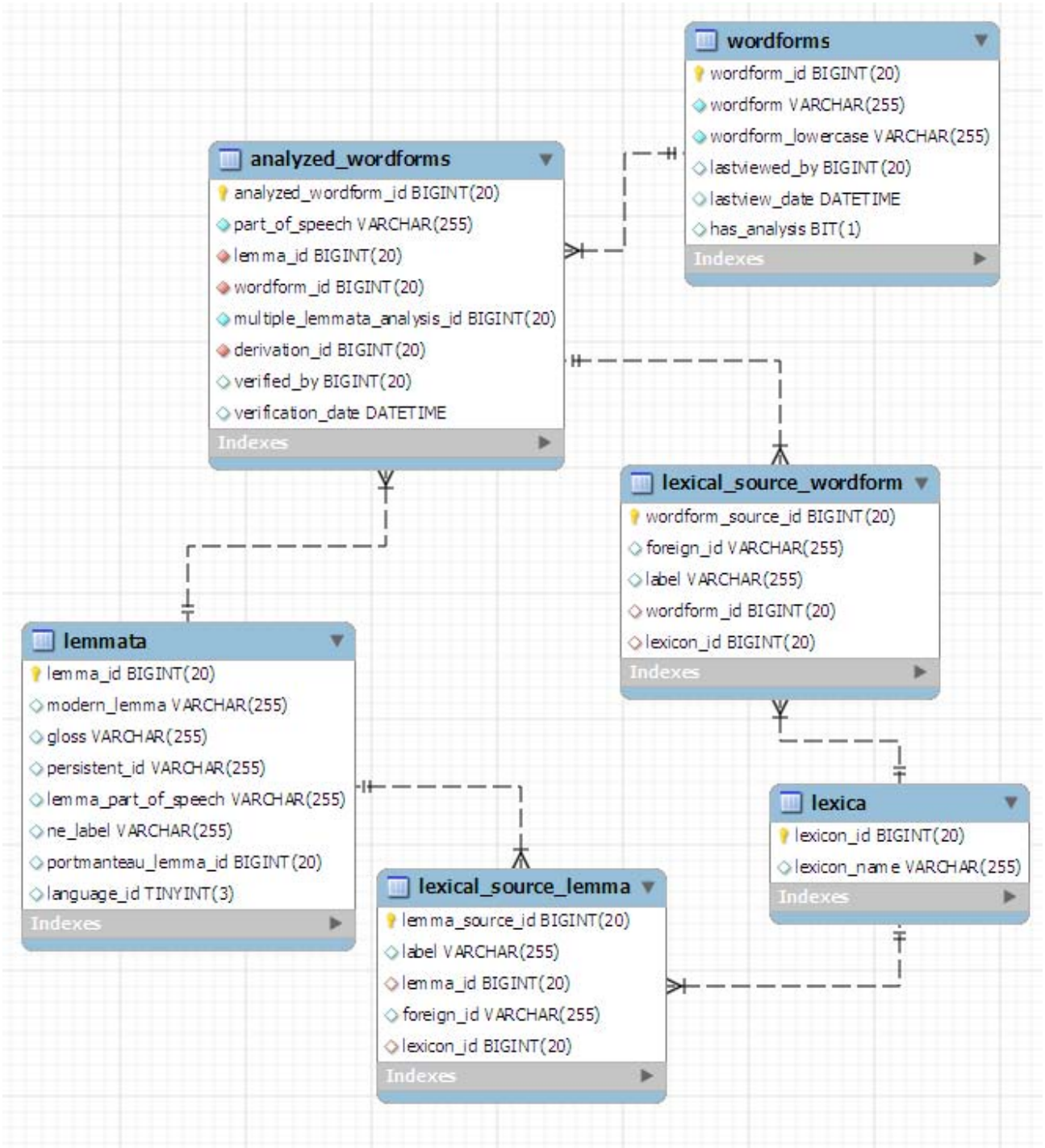


Figure 11. Lexical source

Table lemmata

lemma_id	Modern_lemma	lemma_pos	ne_label
L202	Jan van de Wiel	NOU	NE_PER

Table lexical_source_lemma

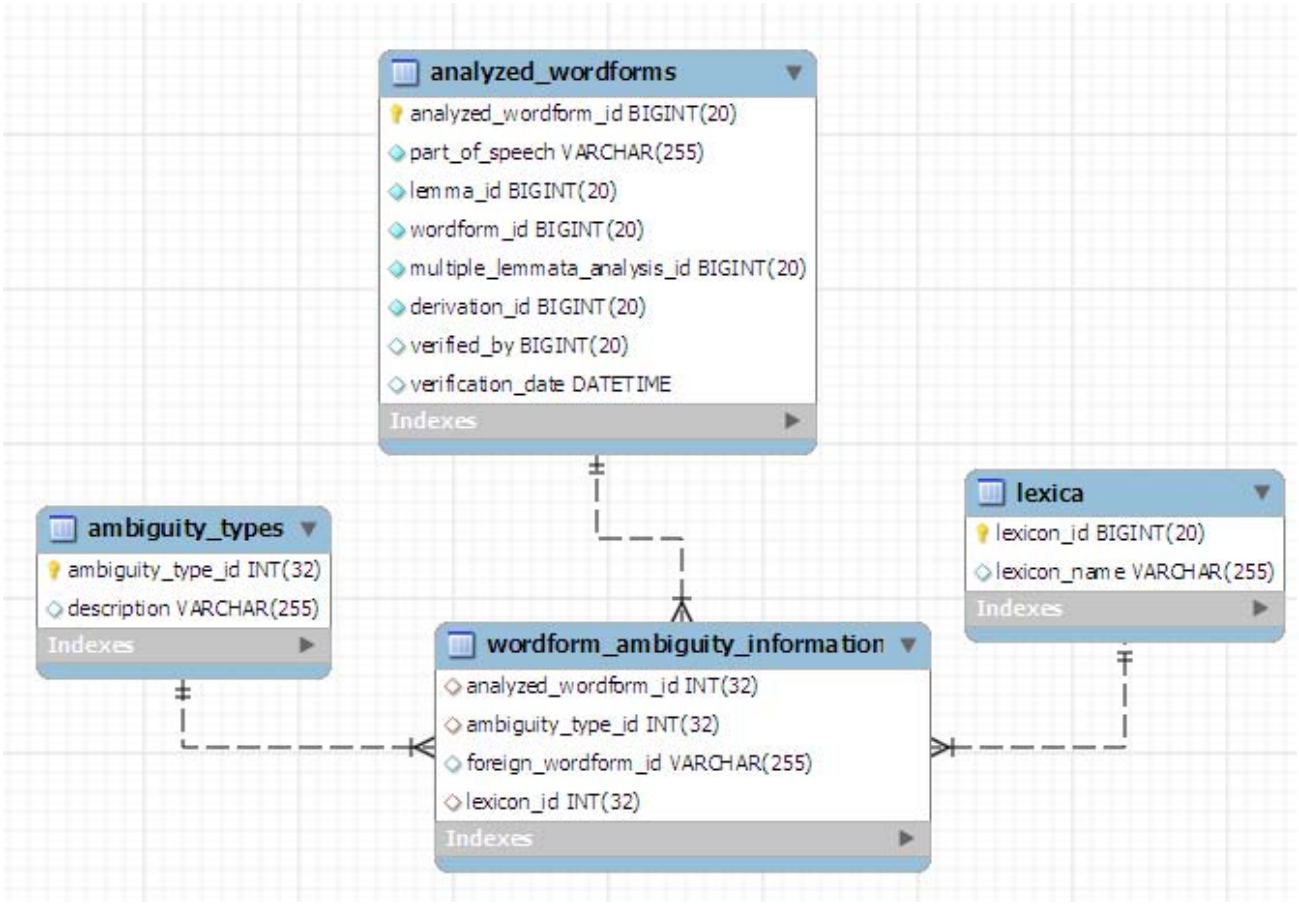
Lemma_source_id	Labels	Lemma_id	Foreign_id	Lexicon_id
Ls1	Physics, Science	A1	0000330x	Lex1

Example: (word index of Van der Sijs)

woonachtig* wonende 1279 [CG 11, 423]
woord* klank met eigen betekenis 776-880 [CG 11 1 Utr. Doopbelofte] {2.5}
woordenboek* dictionaire [Toll.]
worcestersaus kruidige saus 1900 [Sanders 1995] < Engels {4.1.6}
This word list gives us dates of occurrence which can be useful. The information is linked on the lemma level.

Status of lexical source information: optional, internal

6.1. Ambiguity information



Especially for named entities, the information that a word form is also part of the general lexicon or of another part of the named entity lexicon can be useful. Hence, we added some structure to indicate ambiguity of a word form. This ambiguity information may derive from another lexicon or from manual inspection.

7. Converting the database into LMF

7.1. Introduction.

In the previous chapters we described the structure of the database that is used for building the lexicon. The final form of the lexicon, however, will be in the Lexical Markup Framework (LMF: ISO 24613:2008) for this is the standard for sharing computational lexicons.
In this chapter we first describe the structures in LMF that correspond with those that have been discussed above in the format of a relational database.
Second, we will describe the method to compile a LMF-version in XML-format from the relationall database. The scripts that are required for this process and the instructions are provided in Appendix [?].

7.2. Mappings

7.2.1. On notation

The ISO standard uses UML diagrams to represent LMF models. We will do the same in this document. For convenience we will describe the most essential elements of these diagrams. The boxes in Figure 12 represent elements in the XML-structure. Above the line is the name of the element, and below the names of the attributes.

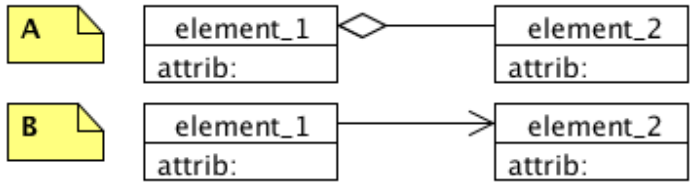


Figure 12. Notation in UML

The arrow in Diagram A indicates that Element 2 is an aggregate of Element 1. Implemented in XML, this means that Element 2 is embedded in Element 1.
The arrow in Diagram B indicates that Element 1 and 2 are associated and that Element 1 can send messages to Element 2. Implemented in XML this means that Element 1 contains a pointer to Element 2.

7.2.2. Unlabelled word forms.

Unlabeled word forms have no linguistic data attached to them.

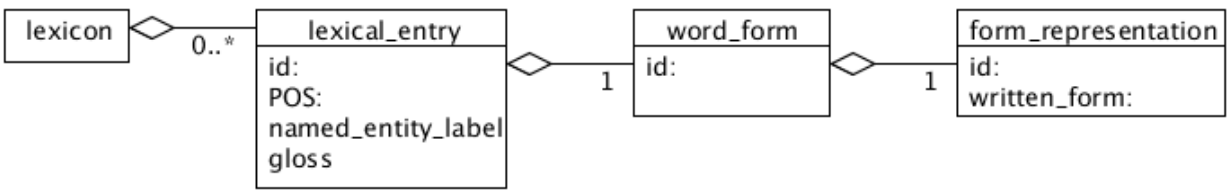


Figure 13. Unlabelled word forms.

The lexicon element is the top-node in our description. The lexical entry (LE) corresponds with what in the previous chapter has been labelled ‘lemma’. The LMF element ‘word form’ corresponds with

the notion 'analyzed wordform' of the previous chapters. And the LMF element 'form_representation' finally, corresponds with the notion 'wordform' of the previous chapters.
In case of the unlabelled word forms, the embedding elements 'lexical_entry' and 'word_form' will contain no linguistic information.

7.2.3. Inflection (labelled word forms).

Labelled word forms have linguistic information attached to them. Information about the available set of features is provided at the level of the LE. The features and values of word forms point to the relevant features that reside under the lexical entry.

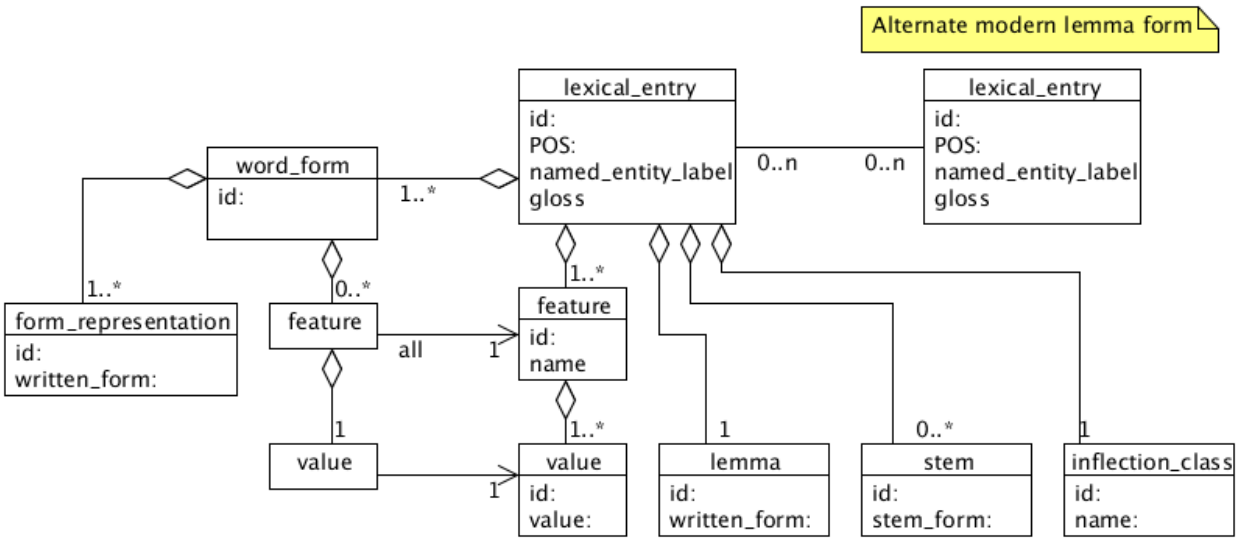


Figure 14. Inflection.

Note that usage of the term 'Lemma' in LMF is differently from that in the previous chapters. In LMF it contains a marker for the LE; usually the stem or base of the word. In this document the element lemma contains the form of the modern 'lemma'.

7.2.4. Composition.

The set of morphological patterns are attached to the level of the lexicon. LE's can have several analyses, which all point to different morphological patterns.

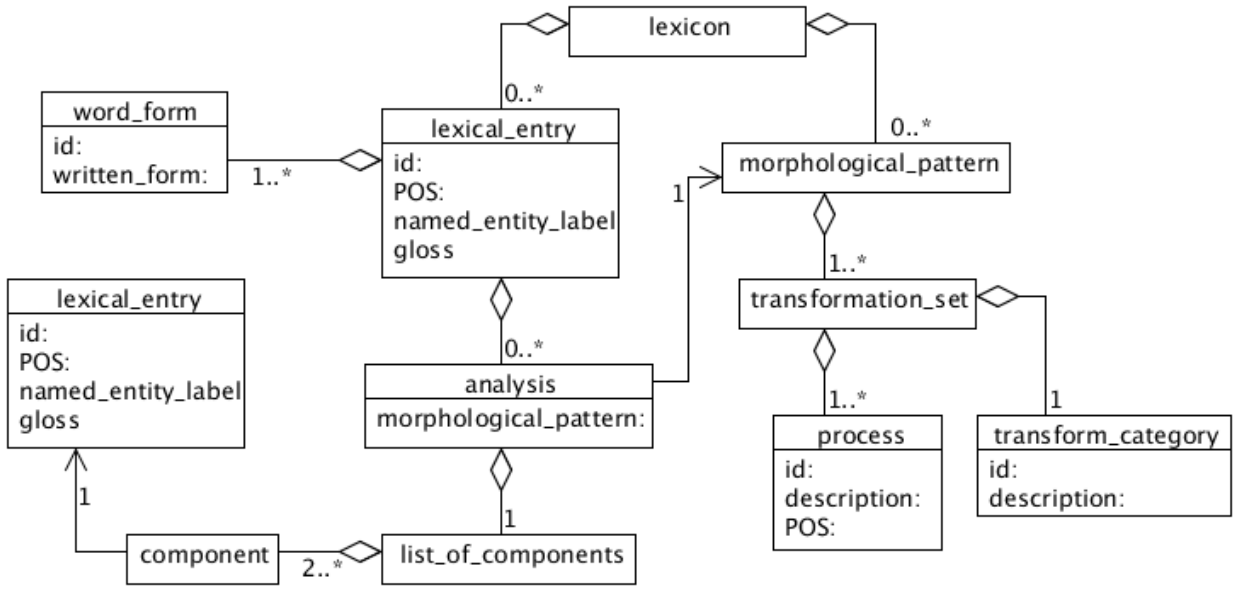


Figure 15. Composition.

7.2.5. Spelling.

It is possible to specify the normalized spelling of word forms in a different (older) spelling.

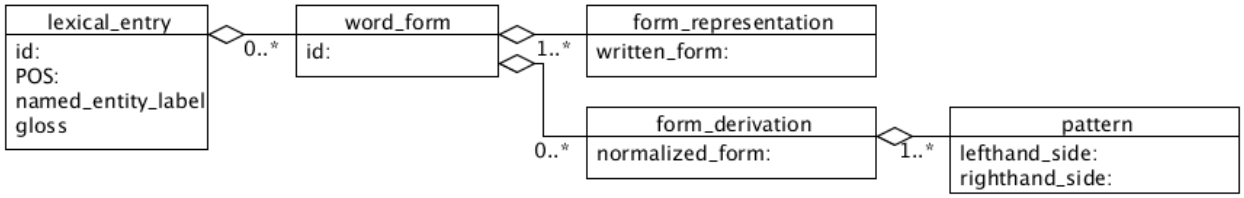


Figure 16. Normalized spelling.

The patterns describe how the written form is derived from the normalized form.

7.2.6. Clitics.

Clitic combination have a lot in common with composition. Both are considered agglutinations. Clitics, however, are analyzed at the level of the word form.

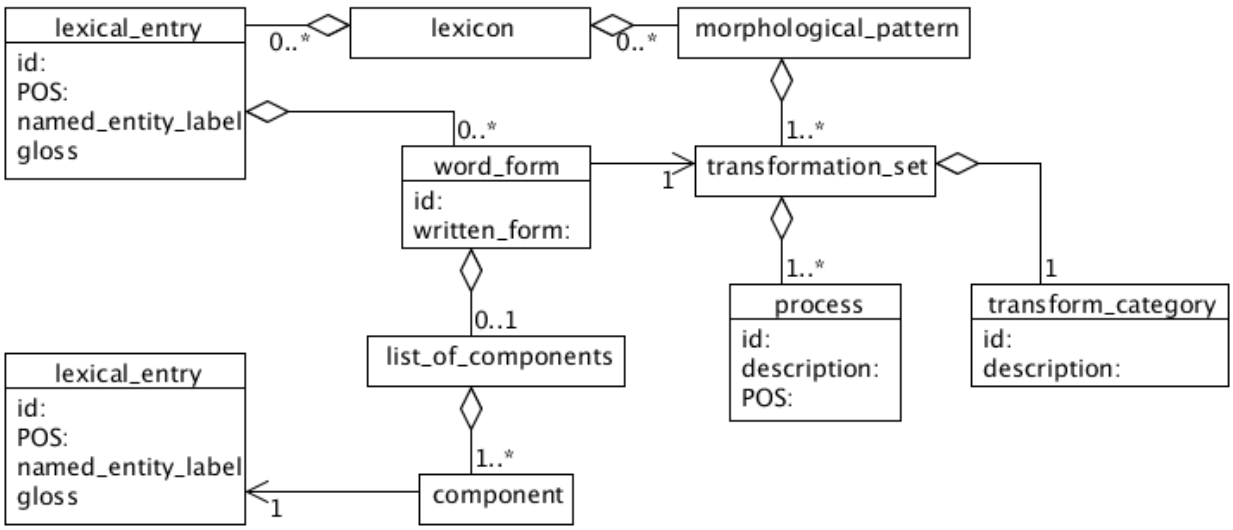


Figure 17. Clitics.

Clitics are represented as LE's that have an aggregated word form, and an ordered list of components. Component are references to other LE's.

7.2.7. Portmanteau.

A portmanteau specifies a relations between homograph lemma's (implemented as LE's). It has been implemented in LMF-format as a lexical entry containing a list of members. Each member in the list points to a lexical entry. The concept of a 'List of Members' is derived from the 'List of Components' that is used for e.g. composition and MWE's. The main difference is that a 'List of Components' is an ordered set and a 'List of members' is not ordered.

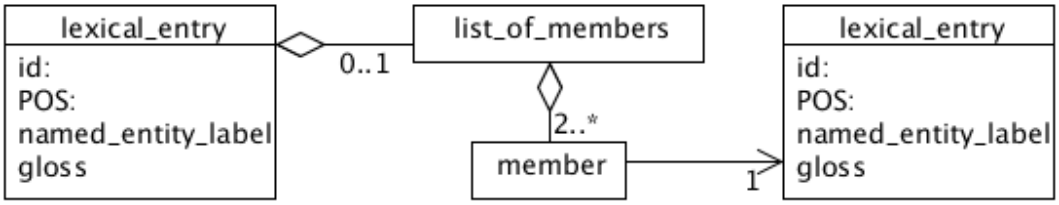


Figure 18. Portmanteau.

7.2.8. Transcategorization.

Transcategorisations specify homonym word forms from different LE's that typically differ in part-of-speech type. Since there is usually a limited list of transcategorisation types in a language, this list is located at the lexicon level.

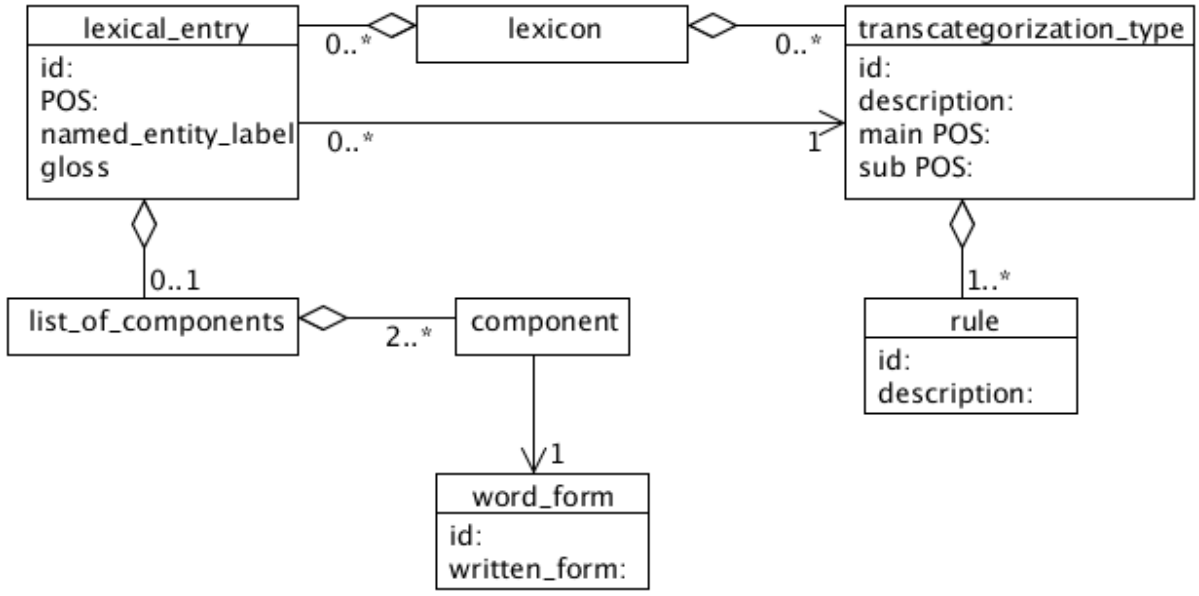


Figure 19. Transcategorization.

For transcategorisations we use Lexical Entries of the type "Categorisation", that point to the according "Transcategorisation Type" and rule. Further, the Lexical Entry contains a 'List of Components' to specify the elements of the transcategorisation. The reason why we do not use a list of members as in the case of Portmanteau's is that a List of Components is ordered.

7.2.9. Multiword expressions.

Multiword expressions are added to the lexicon as Lexical
Entries. The analysis of that LE points to a Multiword Expression Pattern (MWE Pattern). These MWE Patterns describe a ordered list of nodes, and in the description field the grammatical relation of these nodes.

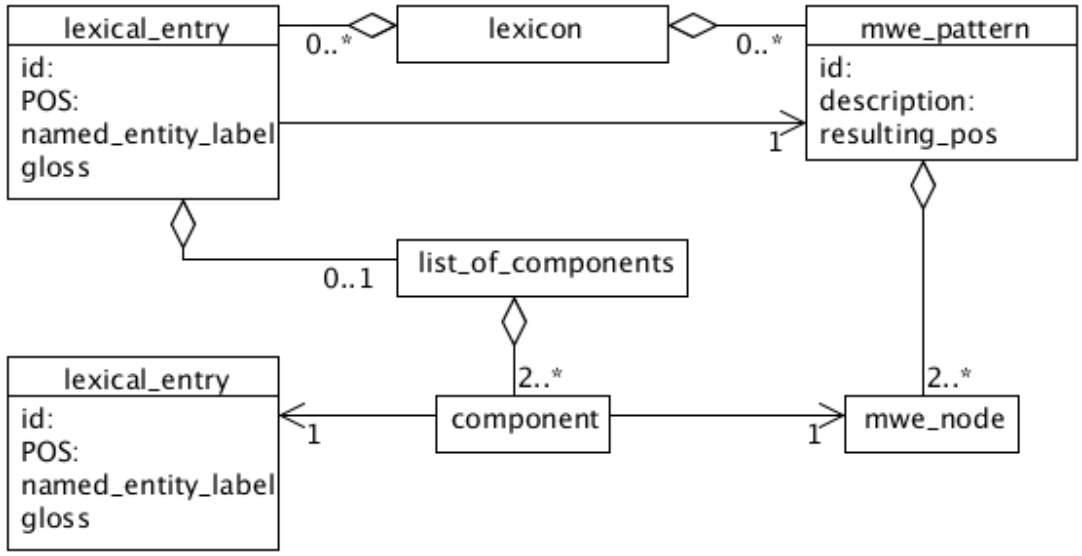


Figure 20. Multiword expressions.

7.2.10. Multiword named entities.

Multiword expressions are added to the lexicon as Lexical
Entries. The analysis of that LE points to a Multiword Expression Pattern (MWE Pattern). These MWE Patterns describe a ordered list of nodes, and in the description field the grammatical relation of these nodes.

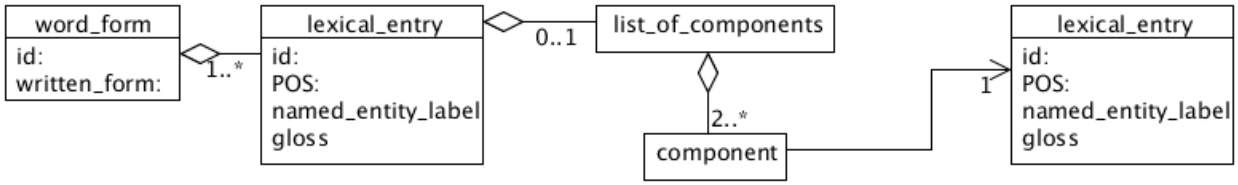


Figure 21. Multiword named entities.

7.2.11. Attestations.

LMF provides a structure for examples of use of a LE. This structure ('context') is subsumed under the 'sense' part of the LE. This is not fit for our purpose since we want the description of the context to clarify the provenance of word forms.
We, therefore, have to create a new extension with new categories for this purpose.

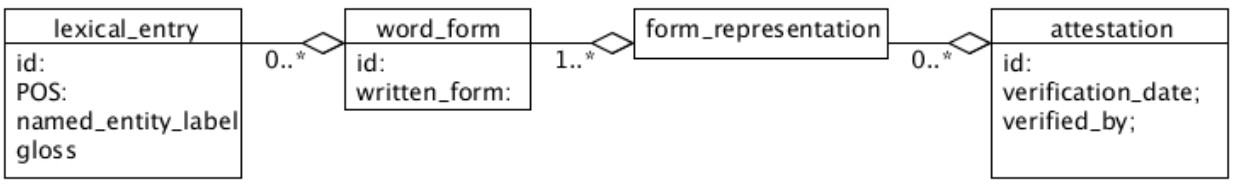


Figure 22. Attestations.

In paragraph 2.5 we described three types of attestations. The types ‘text attestation’ and ‘token attestation’ are attached to the form representations of analyzed word forms. The attestations of unanalyzed word forms are located at the same position, but occur in lexical entries with unlabelled word forms (see par. 7.2.2).

7.3. Converting relational data to XML.

In the previous section we described the LMF-format in XML we want to use for the final form of the lexicon. In this section we present a method for converting the content of the relational database into XML. In Appendix [?] you will find the Perl script (‘relDB2xml.pl’) that can be used for this. The script is to be used in combination with a structure definition for a certain lexicon (language). Appendix [?] contains the specification for the Dutch lexicon (NL_Structure.pl’).

The script 'relDB2xml.pl' is run without arguments. All specific data for the conversion are in a separate (Perl) file which contains the mapping of tables to xml. The file also contains all details on the database that contains the relational data. The reference to this file is specified somewhere at the top of the script 'relDB2xml.pl'.
The mapping specification is laid down in a array structure. Note that this is Perl-code and that using the right syntax is very important.

The array has an embedded structure that roughly corresponds with the resulting xml.
There are three kinds of substructures: for tables, fields and XML elements.

Structure for XML elements.
The arrays for binding contain the following elements:
- element name REQUIRED
- list of arrays for subelements REQUIRED

This structure introduces an XML element which will contain all further data from its subelements.

Structure for tables.
Every array for entering tables contains these elements:
- connection type ("->") REQUIRED
- selection criterium REQUIRED
- name of resulting XML element (can be emtpy string) REQUIRED
- table name REQUIRED
- list of arrays with subelement OPTIONAL

The selection criterium is essentialy the 'where' clause in a SQL select statement. The name of the resulting XML element is used to specify the resulting XML subtree.
If the name is an empty string, no new elements will be introduced at that level, which means that the fields of all records that result from the query will be siblings. If a simple name is specified, a subelement with that name is introduced for every record that results from the query in which the fields of that record are embedded. If a path is specified, (e.g. "element_a.element_b"), extra levels of subelements will be introduced for every record.

Note that there are two ways to introduce XML substructures: using the 'Structure for XML elements' (example A), or using a path specification in the 'Structure for tables' (example B).

```
A:   ["collection",   ["->",   "lemmata.lemma_id=lexical_source.lemma_id",
"source", "lexical_source_lemma"]]
B: ["->", "lemmata.lemma_id=lexical_source.lemma_id", "collection.source",
"lexical_source_lemma"]
```

These will result in different structures when there are more than one record found in the query:

```
A: <collection>
     <source><.. content of record 1 ..></source>
     <source><.. content of record 2 ..></source>
   </collection>

B: <collection>
     <source><.. content of record 1 ..></source>
   </collection>
   <collection>
     <source><.. content of record 2 ..></source>
   </collection>
```

Structure for fields.
Every array for adding fields contains these elements:
- connection type ("-") REQUIRED
- element name REQUIRED
- field name REQUIRED

The element name is the name of the XML element which will hold the value of the field specified by the field name. The element name cannot be an empty string. The element name can be a path, in which case extra levels of XML elements will be introduced (analogue to the examples presented above).
The field name specifies the field that contain the value that has to be inserted into the XML.

8. References

D. Archer, A. Ernst-Gerlach, S. Kempken, Th. Pilz and P. Rayson (2006). The identification of spelling variants in English and German historical texts: manual or automatic?. In *Digital Humanities* (proceedings), Paris, 2006, pp. 3 - 5.

Bień, Janusz S. (2004) *An Approach to Computational Morphology*. In: Intelligent Information Processing and Web Mining. Proceedings of the International IIS:IIP WM'04 Conference held in Zakopane, Poland, May 17-20,2004. Springer, Berlin Heidelberg New York, pp. 181-199. ISBN 3-540-21331-7

S. Cucerzan and D. Yarowsky, Bootstrapping a multilingual part-of-speech tagger in one person-day. In: Dan Roth and Antal van den Bosch (eds.), *Proceedings of CoNLL-2002*, Taipei, Taiwan, 2002, pp. 132-138.

A. Ernst-Gerlach and N. Fuhr. Generating Search Term Variants for Text Collections with Historic Spellings. In *ECIR*, 2006, pp. 49-60.

G. Francopoulo, N. Bel, M. George, N. Calzolari, M. Monachini, M. Pet M.and C. Soria. Lexical Markup Framework: ISO standard for semantic information in NLP lexicons. *GLDV (Gesellschaft für linguistische Datenverarbeitung)*, Tübingen, 2007.

G. Francopoulo, M. George, N. Calzolari, M. Monachini M., N. Bel., M. Pet and C. Soria. Lexical Markup Framework (LMF). *LREC*, Genoa, 2006.

N. Grégoire. Design and Implementation of a Lexicon of Dutch Multiword Expressions. In: N. Grégoire et al. (eds), *Proceedings of the ACL 2007 Workshop on A Broader Perspective on Multiword Expressions*. Prague, 2007, pp. 17–24.

A. Hauser, M. Heller, E. Leiss, K. U. Schulz and C. Wanzeck. Information Access to Historical Documents from the Early New High German Period. In: *IJCAI-2007 Workshop on Analytics for Noisy Unstructured Text Data*, Hyderabad, India - January 8, 2007, pp. 147-154.

V. Hoste, W. Daelemans and S. Gillis, Using rule-induction techniques to model pronunciation variation in Dutch. In: *Computer Speech and Language* 18:1, pp. 1-24.

F. Masini. Multi-word expressions between syntax and the lexicon: The case of Italian verb-particle constructions. In: *SKY Journal of Linguistics* 18 (2005): pp. 145-173.

J.E.J.M. Odijk. A Proposed Standard for the Lexical Representation of Idioms. In: *Proceedings of Euralex*. Lorient, 2004, pp. 153-163.

A. Rappoport, Ari and T. Levent-Levi, Induction of Cross-Language Affix and Letter Sequence Correspondence. In: Proceedings, *EACL 2006 Workshop on Cross-Language Knowledge Induction*, April 2006, Trento, Italy.

E.S. Ristad and P.B. Yianilos. Learning string edit distance. In: *Machine Learning: Proceedings of the Fourteenth International Conference* (San Francisco, July 8-11 1997), D. Fisher, Ed., Morgan Kaufmann, 1997, pp. 287--295.

N. van der Sijs. *Etymologie in het digitale tijdperk, Een chronologisch woordenboek als praktijkvoorbeeld*. Leiden, 2001.

Appendix A: Database schema

Table alternate_modern_lemmata

Field	Type	Null	Key	Default	Extra
alternate_lemma_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
alternate_lemma	varchar(255)	YES		NULL	
base_lemma_id	bigint(20) unsigned	YES	MUL	NULL	

Table analyzed_wordforms

Field	Type	Null	Key	Default	Extra
analyzed_wordform_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
part_of_speech	varchar(255)	NO	MUL		
lemma_id	bigint(20) unsigned	NO	MUL		
wordform_id	bigint(20) unsigned	NO	MUL		
multiple_lemmata_analysis_id	bigint(20) unsigned	NO			
derivation_id	bigint(20) unsigned	NO	MUL		
verified_by	bigint(20) unsigned	YES		NULL	
verification_date	datetime	YES		NULL	

Table conversion_rules

Field	Type	Null	Key	Default	Extra
rule_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
main_pos	varchar(255)	YES		NULL	
sub_pos	varchar(255)	YES		NULL	
transcategorization_id	bigint(20) unsigned	YES	MUL	NULL	

Table corpora

Field	Type	Null	Key	Default	Extra
corpus_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
name	varchar(255)	YES		NULL	

Table corpusId_x_documentId

Field	Type	Null	Key	Default	Extra
corpus_id	bigint(20) unsigned	NO	PRI		
document_id	bigint(20) unsigned	NO	PRI		

Table derivations

Field	Type	Null	Key	Default	Extra
derivation_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
normalized_form	varchar(255)	YES	MUL	NULL	
pattern_application_id	bigint(20) unsigned	NO			

Table documents

Field	Type	Null	Key	Default	Extra

document_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
persistent_id	varchar(255)	YES	MUL	NULL	
word_count	bigint(20) unsigned	YES		NULL	
encoding	bigint(20) unsigned	YES		NULL	
title	varchar(255)	YES		NULL	
year_from	bigint(20) unsigned	YES		NULL	
year_to	bigint(20) unsigned	YES		NULL	
pub_year	bigint(20) unsigned	YES		NULL	
author	varchar(255)	YES		NULL	
editor	varchar(255)	YES		NULL	
publisher	varchar(255)	YES		NULL	
publishing_location	varchar(255)	YES		NULL	
text_type	varchar(255)	YES		NULL	
region	varchar(255)	YES		NULL	
language	varchar(255)	YES		NULL	
other_languages	varchar(255)	YES		NULL	
spelling	varchar(255)	YES		NULL	
parent_document	bigint(20) unsigned	YES	MUL	NULL	

Table dont_show

Field	Type	Null	Key	Default	Extra
wordform_id	bigint(20) unsigned	NO	PRI		
document_id	bigint(20) unsigned	NO	PRI	0	
corpus_id	bigint(20) unsigned	NO	PRI	0	
at_all	tinyint(3) unsigned	NO	PRI	0	
user_id	bigint(20) unsigned	NO			
date	datetime	NO			

Table group_attestations

Field	Type	Null	Key	Default	Extra
group_attestation_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
token_id	bigint(20) unsigned	YES		NULL	
quote	text	YES		NULL	
analyzed_wordform_id	bigint(20) unsigned	NO	MUL		
derivation_id	bigint(20) unsigned	NO			
wordform_group_id	bigint(20) unsigned	NO			

Table inflection_classes

Field	Type	Null	Key	Default	Extra
inflection_class_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
inflection_class_name	varchar(255)	YES		NULL	

Table languages

Field	Type	Null	Key	Default	Extra

language_id	tinyint(3) unsigned	NO	PRI	NULL	auto_increment
language	varchar(255)	NO	UNI		

Table lemma_feature_assignments

Field	Type	Null	Key	Default	Extra
assignment_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
feature_id	bigint(20) unsigned	YES	MUL	NULL	
value_id	bigint(20) unsigned	YES	MUL	NULL	
lemma_id	bigint(20) unsigned	YES	MUL	NULL	

Table lemma_feature_values

Field	Type	Null	Key	Default	Extra
lemma_feature_value_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
lemma_feature_value	varchar(255)	YES		NULL	

Table lemma_features

Field	Type	Null	Key	Default	Extra
lemma_feature_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
lemma_feature_name	varchar(255)	YES		NULL	

Table lemma_inflection_class

Field	Type	Null	Key	Default	Extra
lemma_inflection_class_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
lemma_id	bigint(20) unsigned	YES	MUL	NULL	
inflection_class_id	bigint(20) unsigned	YES	MUL	NULL	

Table lemmata

Field	Type	Null	Key	Default	Extra
lemma_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
modern_lemma	varchar(255)	YES	MUL	NULL	
gloss	varchar(255)	YES		NULL	
persistent_id	varchar(255)	YES		NULL	
lemma_part_of_speech	varchar(255)	YES		NULL	
ne_label	varchar(255)	YES		NULL	
portmanteau_lemma_id	bigint(20) unsigned	YES	MUL	NULL	
language_id	tinyint(3) unsigned	YES		NULL	

Table lexica

Field	Type	Null	Key	Default	Extra
lexicon_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
lexicon_name	varchar(255)	YES		NULL	

Table lexical_source_lemma

Field	Type	Null	Key	Default	Extra
lemma_source_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment

label	varchar(255)	YES		NULL
lemma_id	bigint(20) unsigned	YES	MUL	NULL
foreign_id	varchar(255)	YES		NULL
lexicon_id	bigint(20) unsigned	YES	MUL	NULL

Table lexical_source_wordform

Field	Type	Null	Key	Default	Extra
wordform_source_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
foreign_id	varchar(255)	YES		NULL	
label	varchar(255)	YES		NULL	
wordform_id	bigint(20) unsigned	YES	MUL	NULL	
lexicon_id	bigint(20) unsigned	YES	MUL	NULL	

Table morphological_analyses

Field	Type	Null	Key	Default	Extra
morphological_analysis_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
arity	bigint(20) unsigned	YES		NULL	
analyzed_lemma_id	bigint(20) unsigned	YES	MUL	NULL	
morphological_operation_id	bigint(20) unsigned	YES	MUL	NULL	

Table morphological_operations

Field	Type	Null	Key	Default	Extra
morphological_operation_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
description	varchar(255)	YES		NULL	
resulting_part_of_speech	varchar(255)	YES		NULL	

Table multiple_lemmata_analyses

Field	Type	Null	Key	Default	Extra
multiple_lemmata_analysis_id	bigint(20) unsigned	NO	PRI		
multiple_lemmata_analysis_part_id	bigint(20) unsigned	NO	PRI		
part_number	bigint(20) unsigned	NO	PRI		
nr_of_parts	tinyint(3) unsigned	NO	PRI		

Table multiple_lemmata_analysis_parts

Field	Type	Null	Key	Default	Extra
multiple_lemmata_analysis_part_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
part_of_speech	varchar(255)	NO	MUL		
lemma_id	bigint(20) unsigned	NO			

Table multiword_analyses

Field	Type	Null	Key	Default	Extra
multiword_analysis_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
arity	bigint(20) unsigned	YES		NULL	
analyzed_lemma_id	bigint(20) unsigned	YES	MUL	NULL	
multiword_operation_id	bigint(20) unsigned	YES	MUL	NULL	

Table multiword_operations

Field	Type	Null	Key	Default	Extra
multiword_operation_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
description	varchar(255)	YES		NULL	
resulting_pos	varchar(255)	YES		NULL	

Table ne_variant_relation_types

Field	Type	Null	Key	Default	Extra
ne_variant_relation_type_id	int(32)	NO	PRI	NULL	auto_increment
ne_variant_relation_name	varchar(255)	YES		NULL	
ne_variant_relation_desciption	text	YES		NULL	

Table ne_variant_relations

Field	Type	Null	Key	Default	Extra
first_lemma_id	int(32)	YES	MUL	NULL	
second_lemma_id	int(32)	YES		NULL	
ne_variant_relation_type_id	int(32)	YES		NULL	

Table paradigm_positions

Field	Type	Null	Key	Default	Extra
paradigm_position_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
paradigm_position_name	varchar(255)	YES		NULL	
paradigm_position	bigint(20) unsigned	YES		NULL	
paradigm_id	bigint(20) unsigned	YES	MUL	NULL	
transformset_id	bigint(20) unsigned	YES	MUL	NULL	

Table paradigms

Field	Type	Null	Key	Default	Extra
paradigm_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
paradigm_name	varchar(255)	YES		NULL	

Table part_morphological_analysis

Field	Type	Null	Key	Default	Extra
part_morphological_analysis_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
part_number	bigint(20) unsigned	YES		NULL	
part_lemma_id	bigint(20) unsigned	YES	MUL	NULL	
morphological_analysis_id	bigint(20) unsigned	YES	MUL	NULL	

Table part_multiword_analysis

Field	Type	Null	Key	Default	Extra
part_multiword_analysis_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
part_number	bigint(20) unsigned	YES		NULL	
part_lemma_id	bigint(20) unsigned	YES	MUL	NULL	
multiword_analysis_id	bigint(20) unsigned	YES	MUL	NULL	

Table pattern_applications

Field	Type	Null	Key	Default	Extra
pattern_application_id	bigint(20) unsigned	NO	MUL		
position	bigint(20) unsigned	YES		NULL	
pattern_id	bigint(20) unsigned	YES		NULL	
number_of_patterns	bigint(20) unsigned	NO			

Table patterns

Field	Type	Null	Key	Default	Extra
pattern_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
left_hand_side	varchar(64)	YES	MUL	NULL	
right_hand_side	varchar(64)	YES		NULL	

Table stem_types

Field	Type	Null	Key	Default	Extra
stem_type_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
stem_type_name	varchar(255)	YES		NULL	

Table stems

Field	Type	Null	Key	Default	Extra
stem_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
stem_form	varchar(255)	YES		NULL	
lemma_id	bigint(20) unsigned	YES	MUL	NULL	
stem_type_id	bigint(20) unsigned	YES	MUL	NULL	

Table text_attestation_verifications

Field	Type	Null	Key	Default	Extra
document_id	bigint(20) unsigned	NO	PRI		
wordform_id	bigint(20) unsigned	NO	PRI		
verification_date	datetime	NO			
verified_by	bigint(20) unsigned	NO			

Table text_attestations

Field	Type	Null	Key	Default	Extra
attestation_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
frequency	bigint(20) unsigned	YES		NULL	
analyzed_wordform_id	bigint(20) unsigned	NO	MUL		
document_id	bigint(20) unsigned	NO			

Table token_attestation_verifications

Field	Type	Null	Key	Default	Extra
document_id	bigint(20) unsigned	NO	PRI		
wordform_id	bigint(20) unsigned	NO	PRI		
start_pos	bigint(20) unsigned	NO	PRI		

end_pos	bigint(20) unsigned	NO			
verification_date	datetime	NO			
verified_by	bigint(20) unsigned	NO			

Table token_attestations

Field	Type	Null	Key	Default	Extra
attestation_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
token_id	bigint(20) unsigned	YES		NULL	
quote	text	YES		NULL	
analyzed_wordform_id	bigint(20) unsigned	NO	MUL		
derivation_id	bigint(20)	NO			
document_id	bigint(20) unsigned	NO			
start_pos	bigint(20) unsigned	NO			
end_pos	bigint(20) unsigned	NO			

Table transcategorization_types

Field	Type	Null	Key	Default	Extra
transcategorizationtype_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
description	varchar(255)	YES		NULL	
main_pos	varchar(255)	YES		NULL	
sub_pos	varchar(255)	YES		NULL	

Table transcategorizations

Field	Type	Null	Key	Default	Extra
transcategorization_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
mainlemma_id	bigint(20) unsigned	YES	MUL	NULL	
sublemma_id	bigint(20) unsigned	YES	MUL	NULL	
transcategorizationtype_id	bigint(20) unsigned	YES	MUL	NULL	

Table transformsets

Field	Type	Null	Key	Default	Extra
transformset_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
inflection_process	varchar(255)	YES		NULL	
formal_tag	varchar(255)	YES		NULL	
stem_type_id	bigint(20) unsigned	YES	MUL	NULL	

Table type_frequencies

Field	Type	Null	Key	Default	Extra
type_frequency_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
frequency	bigint(20) unsigned	NO			
wordform_id	bigint(20) unsigned	NO	MUL		
document_id	bigint(20) unsigned	NO			

Table users

Field	Type	Null	Key	Default	Extra

user_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
name	varchar(255)	YES	UNI	NULL	

Table wordform_groups

Field	Type	Null	Key	Default	Extra
wordform_group_id	bigint(20) unsigned	NO	PRI		
document_id	bigint(20) unsigned	NO	PRI		
onset	bigint(20) unsigned	NO	PRI		
offset	bigint(20) unsigned	NO	PRI		

Table wordform_transform_instance

Field	Type	Null	Key	Default	Extra
transform_instance_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
transformset_id	bigint(20) unsigned	YES	MUL	NULL	
stem_id	bigint(20) unsigned	YES	MUL	NULL	
analyzed_wordform_id	bigint(20) unsigned	YES	MUL	NULL	

Table wordforms

Field	Type	Null	Key	Default	Extra
wordform_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
wordform	varchar(255)	NO	UNI		
wordform_lowercase	varchar(255)	NO	MUL		
lastviewed_by	bigint(20)	YES		NULL	
lastview_date	datetime	YES		NULL	
has_analysis	bit(1)	YES		NULL	

Appendix B: Filters for the export of relevant subsets from the lexicon

Filters for various applications will be developed as the workflow for lexicon development and deployment progresses. They can be implemented eithers as SQL queries on the database or, for instance, as XSLT queries on the XML export format.

A simple example: produce a word list with frequencies for all documents from 1749.

```
create view lemma_wordform_attestation as select modern_lemma,lemmata.lemma_id,
wordform, pos, documents.document_id, documents.year_from, documents.year_to,
documents.title,type_attestations.frequency from lemmata, analyzed_wordforms,
wordforms, type_level_attestations, documents where analyzed_wordforms.lemma_id =
lemmata.lemma_id and wordforms.wordform_id = analyzed_wordforms.wordform_id and
type_level_attestations.analyzed_wordform_id=analyzed_wordforms.analyzed_wordform_i
d and type_level_attestations.document_id=documents.document_id;

select distinct wordform, sum(frequency) as frequency from
lemma_wordform_attestation where year_from=1749 and year_to =1749 group by
wordform;
```

Appendix C: Script for converting relational data to LMF (XML):'relDB2xml.pl'.

The script writes output to 'STDOUT'. After the keyword 'require', the name of the file containing the structure and further parameters has to be provided.

```
use strict;
use DBI;
use HTML::Entities;

# Every array for inserting tables contains these elements:
# - connection type ("->") REQUIRED
# - selection criteria REQUIRED
# - element name (can be emtpy string) REQUIRED
# - table name REQUIRED
# - list of arrays with subelement OPTIONAL

# Every array for inserting fields contains these elements:
# - connection type ("-") REQUIRED
# - element name (can be emtpy string) REQUIRED
# - table name REQUIRED

# arrays for binding contain the following elements:
# - element name REQUIRED
# - list of arrays for subelements REQUIRED

require "NL_Structure.pl";

open (LOG, sprintf ">%s.log", getParam ("output"));
```

```
my $dbh = DBI->connect (sprintf ("DBI:mysql:database=%s;host=%s", getParam
("database"),  getParam  ("databasehost")),  getParam  ("user"),  getParam
("password"));
if (!defined ($dbh)) {
    die sprintf "Unable to connect: %s\n", $DBI::errstr;
}

printf "%s\n",  xmlHeader (getParam ("dtd"));
printf "%s", buildXml ("", @{getLmf()});

$dbh->disconnect;

close (LOG);

sub buildXml {
    my ($super, $type, @rest) = @_;
    if (@rest) {
      if ($type eq "->") { # handle table
          my ($constraint, $tag, $table) = splice (@rest, 0, 3);
          my @table = queryAggregate ($dbh, $super, $constraint, $table);
          my ($result, $openTag, $closeTag) = ("", "", "");
          if ($tag ne "") {
            $openTag = "<" . $tag . ">";
            $closeTag = "</" . $tag . ">";
          }
          foreach my $record (@table) {
            $result .= sprintf "%s%s%s\n", $openTag, join ("", map {buildXml
($record, @{$_})} @rest), $closeTag;
          }
          return $result;
      }
      elsif ($type eq "-") { #handle field
          my ($name, $key) = @rest;
          my @path = split (/\./, $name);
          return sprintf "<%s>%s</%s>\n", join ("><", @path), $$super{$key},
join ("></", reverse @path);
      }
      else { #binding element
          if ($type =~ s!^([^.]+)\.!!) {
            return  sprintf  "<%s>\n%s</%s>\n",  $1,  buildXml  ($super,  $type,
@rest), $1;
          }
          else {
            return sprintf "<%s>\n%s</%s>\n", $type, join ("", map {buildXml
($super, @{$_})} @rest), $type;
          }
      }
    }
}


sub queryAggregate {
    my ($dbh, $super, $constraint, $table) = @_;
    my $sth = "";
```

```
    if ($constraint ne "") {
     my ($leftTable, $leftKey, $rightTable, $rightKey) = split (/[.=]/,
$constraint);
     my $query = sprintf "select * from %s where %s='%s'", $rightTable,
$rightKey, $$super{$leftKey};
     $sth = $dbh->prepare ($query);
    }
    else {
     my $query = sprintf "select * from %s", $table;
     $sth = $dbh->prepare ($query);
    }
    $sth->execute or printf LOG "%s\n", $sth->errstr;
    my @result = ();
    my $hashref = "";
    while ($hashref = $sth->fetchrow_hashref) {
     push (@result, $hashref);
    }
    return @result;
}

sub xmlHeader {
    my ($name) = @_;
    if ($name ne "") {
     return sprintf "<?xml version='1.0'?>\n<!DOCTYPE lexicon SYSTEM
'%s'>\n", $name;
    }
    else {
     return "<?xml version='1.0'?>\n";
    }
}
```

Appendix D: Structure Definition for the Dutch Lexicon.

The file contains Perl code. Two data structures are specified: a hash with some details for connecting to a relational database. The parameter 'output' is used to provide a name for the log file. The keyword 'dtd' is optional.

The file further contains two small functions needed to pass the data to the main script. These should not be changed.

```
use strict;

my %params =
("output" => "NL_Lexicon",
 "database" => "EE3",
 "databasehost" => "impactdb.inl.loc",
 "password" => "impact",
 "user" => "impact",
 "dtd" => "NL_Structure.dtd"
 );

my $lmf =
    ["lexicon",
# rule section
```

```
     ["->", "", "lemma_feature", "lemma_features"],
#     ["->", "", "lemma_feature_value", "lemma_feature_values"],
     ["->", "", "inflection_class", "inflection_classes"],
     ["->", "", "derivation_pattern", "patterns"],
     ["->", "", "transcategorization_type", "transcategorization_types",
      ["->",
"transcategorization_types.transcategorizationtype_id=conversion_rules.tran
scategorization_id", "rule", "conversion_rules"]
     ],
     ["->", "", "mwe_pattern", "multiword_operations",
      ["-", "multiword_operation_id", "multiword_operation_id"],
      ["-", "description", "description"],
      ["-", "resulting_pos", "resulting_pos"],
     ],
     ["->",     "",     "morphological_pattern.transformation_set.process",
"morphological_operations"],
     ["->",         "",         "morphological_pattern.transformation_set",
"transformsets",
      ["->",          "transformsets.stem_type_id=stem_types.stem_type_id",
"transform_category", "stem_types"],
      ["->",
"transformsets.paradigm_position_name=paradigm_positions.paradigm_position_
name", "process", "paradigm_positions",
       ["->",       "paradigm_positions.paradigm_id=paradigms.paradigm_id",
"paradigm", "paradigms"],
      ],
     ],
# lexical entries
     ["->", "", "lexical_entry", "multiword_analyses",
      ["-", "multiword_analysis_id"],
      ["-", "multiword_operation_id", "mwe_pattern"],
      ["-", "arity", "arity"],
      ["list_of_components",
       ["->",
"multiword_analyses.multiword_analysis_id=part_multiword_analysis.multiword
_analysis_id", "component", "part_multiword_analysis",
      ["-", "part_number", "part_number"],
      ["-", "lemma_id", "part_lemma_id"]
       ]
      ],
     ],
     ["->", "", "lexical_entry", "transcategorizations",
      ["-", "", "transcategorization_type"],
      ["list_of_components",
       ["-", "component.mainlemma_id", "mainlemma_id"],
       ["-", "component.sublemma_id", "sublemma_id"],
      ]
     ],
     ["->", "", "lexical_entry", "lemmata",
      ["-", "lemma_id", "lemma_id"],
      ["-", "modern_lemma", "modern_lemma"],
      ["-", "gloss", "gloss"],
      ["-", "POS", "lemma_part_of_speech"],
      ["-", "ne_label", "ne_label"],
```

```
      ["-", "language_id", "language_id"],
      ["-", "portmanteau_lemma_id", "portmanteau_lemma_id"],
      ["->",      "lemmata.lemma_id=alternate_modern_lemmata.base_lemma_id",
"alternate_modern_lemma", "alternate_modern_lemmata",
       ["-", "alternate_lemma", "alternate_lemma"],
      ],
      ["->",             "lemmata.lemma_id=lemma_inflection_class.lemma_id",
"inflection_class", "lemma_inflection_class",
       ["-", "inflection_class_id", "inflection_class_id"],
      ],
      ["->",   "lemmata.lemma_id=lexical_source_lemma.lemma_id",   "source",
"lexical_source_lemma",
       ["-", "label", "label"],
       ["-", "foreign_id", "foreign_id"],
       ["-", "lexicon_id", "lexicon_id"],
      ],
      ["->", "lemmata.lemma_id=stems.lemma_id", "stem", "stems",
       ["-", "stem_form", "stem_form"],
       ["-", "stem_id", "stem_id"],
       ["->",        "stems.stem_type_id=stem_types.stem_type_id",       "",
"stem_types",
     ["-", "name", "stem_type_name"],
       ],
      ],
      ["->",          "lemmata.lemma_id=lemma_feature_assignments.lemma_id",
"feature", "lemma_feature_assignments",
       ["->",
"lemma_feature_assignments.feature_id=lemma_features.lemma_feature_id",  "",
"lemma_features",
        ["-", "feature_id", "feature_id"],
     ["-", "name", "lemma_feature_name"],
       ],
       ["->",
"lemma_feature_assignments.value_id=lemma_feature_values.lemma_feature_valu
e_id", "value", "lemma_feature_values",
        ["-", "value_id", "lemma_feature_value_id"],
        ["-", "value", "lemma_feature_value"],
       ]
      ],
      ["->",    "lemmata.lemma_id=morphological_analyses.analyzed_lemma_id",
"analysis", "morphological_analyses",
       ["-", "morphological_operation_id", "morphological_operation_id"],
       ["list_of_components",
     ["->",
"morphological_analyses.morphological_analysis_id=part_morphological_analys
is.morphological_analysis_id", "component", "part_morphological_analysis",
       ["-", "number", "part_number"],
       ["-", "lemma_id", "part_lemma_id"],
     ]
       ]
      ],
      ["->",   "lemmata.lemma_id=analyzed_wordforms.lemma_id",   "wordform",
"analyzed_wordforms",
       ["->",  "analyzed_wordforms.derivation_id=derivations.derivation_id",
```

```
"", "derivations",
      ["pattern",
       ["->",
"derivations.derivation_id=pattern_applications.derivation_id",          "",
"pattern_applications",
        ["-", "position", ""],
        ["->",   "pattern_applications.pattern_id=patterns.pattern_id",   "",
"pattern",
         ["-", "left_hand_side", "left_hand_side"],
         ["-", "right_hand_side", "right_hand_side"],
        ]
       ]
      ]
        ],
        ["->",
"analyzed_wordforms.wordform_id=lexical_source_wordform.wordform_id",
"source", "lexical_source_wordform"],
       ["form_representation",
      ["->",   "analyzed_wordforms.wordform_id=wordforms.wordform_id",   "",
"wordforms",
       ["-", "wordform_id", "wordform_id"],
       ["-", "written_form", "wordform"],
      ],
      ["->",
"wordforms.analyzed_wordform_id=text_attestations.analyzed_wordform_id",
"attestation", "text_attestations",
       ["-", "id", "attestation_id"],
       ["-", "frequency", "frequency"],
       ["-", "document_id", "document_id"],
      ],
      ["->",
"analyzed_wordforms.analyzed_wordform_id=token_attestations.analyzed_wordfo
rm_id", "attestation", "token_attestations",
       ["-", "id", "attestation_id"],
       ["-", "token_id", "token_id"],
       ["-", "quote", "quote"],
       ["-", "derivation_id", "derivation_id"],
       ["-", "document_id", "document_id"],
       ["-", "start_pos", "start_pos"],
       ["-", "end_pos", "end_pos"],
      ],
        ],
        ["->",
"analyzed_wordforms.analyzed_wordform_id=wordform_transform_instance.analyz
ed_wordform_id", "", "wordform_transform_instance"]
       ]
      ]
     ];

sub getLmf {
    return $lmf;
}

sub getParam {
```

```
    my ($key) = @_;
    return $params{$key};
}
```