The Representation of Lexical Objects[1]

Erik-Jan van der Linden, Sjaak Brinkkemper, Koenraad De Smedt, Pauline van Boven, and Mieke van der Linden[2]

Summary

Sophisticated applications in the area of language technology require detailed lexical knowledge. To describe the knowledge in a lexical information system adequately, analysis methods from computer science can be used. These methods deliver an abstract and concise description of the system and of the lexical objects, and reveal the considerations that are used when establishing the identity of the objects in the lexicon. Non-linguistic considerations related to the objective of the lexical information system constitute the way in which linguistic factors eventually determine the structure of the lexicon.

Two principles are being introduced, that underlie the lexicon: the *abstraction principle* positing that objects that do not occur in reality have to be represented in the lexicon, and the *generalization principle* stating that the inclusion of these objects necessitates linguistic generalizations in the lexicon.

1. Introduction

In linguistics, and also in computational linguistics, little attention has been paid to the formal description of the lexicon. Only recently, within the paradigms of generative morphology (Scalise 1984) and two-level morphology (Koskenniemi 1983) has interest in lexicology increased. In computational linguistics, this growing interest and the fact that large dictionaries can be represented on computerized media and accessed in real-time have also increased the interest in computational lexicology. Moreover, since systems such as advanced text processing systems and dialogue systems become applicable, a need arises for lexical and morphological components that cover a large part of the lexical data and the lexical and morphological knowledge of a language. The more sophisticated these systems are, the more abstract and detailed knowledge they need (Van der Linden and De Smedt 1987).

For a correct and adequate description of a large quantity of lexical knowledge and the classification thereof in a computer lexicon, a schematic description of the knowledge is necessary. This paper will present a modelling method from computer science, consisting of two stages. In the *system analysis* (paragraph 2.1), an inventory is made of the applications of our so-called 'Lexical Information System' (LIS), a sophisticated lexical component. The *information analysis* (paragraph 2.2) establishes the relations between the information about the lexical objects in this system and the objects themselves (Verheyen and van Bekkum 1982). This second stage reveals considerations that are used when establishing the *identity* of the objects in the lexicon (paragraph 3). Non-linguistic considerations related to the

purpose of the lexical information system constitute the way in which linguistic factors eventually determine the structure of the lexicon (paragraph 4). We end with some conclusions regarding this research.

2. Analysis

2.1 System Analysis

During the *system analysis*, an inventory is made of all applications of the intended information system: e.g. in an advanced test processing software package (Kempen et al. 1987), we distinguish among others:

* modules for hyphenation; these modules use information about stress, pronunciation and the structure of words;
* modules for morphological analysis and generation; these modules use knowledge to analyse words that do not strictly belong to the data in the database;
* modules for syntactic analysis and generation; these modules need syntactic properties that relate to syntactic valence. The components rely partly upon the morphological modules for this knowledge.
* a last component of the system to be mentioned here is a so-called 'linguistic spreadsheet'. When the user of the system changes a property of a word in a text, other words that are related to this word change automatically. If, for instance, the number of the subject is changed from singular to plural the verb changes accordingly. This function uses the syntactic and morphological components of the system.

The modules that use lexical knowledge determine the kind of information stored in the lexicon. Simple applications require simple information, and sophisticated applications require detailed lexical information and therefore a linguistically structured Lexical Information System. This forces to the addition of abstract information: entities must be stored in the lexicon that are abstract in the sense that they don't occur in reality. We name this the *abstraction-principle.*

Abstraction on the level of lexical data and knowledge necessitates the exclusion of linguistic redundancy and the addition of linguistic generalization. We name this the *generalization-principle.*

These principles lead to two charcteristics of the global structure of the lexicon, we will discuss now.

1— Objects in the lexicon are represented as (a) abstract entities notated (b) in a phonetic representation.

2— The lexicon does not have a flat structure, but consists of several layers.

(1a) As an invariant, as stem of an inflectional paradigm, the singular form can be chosen for nouns, the infinitive for verbs, and the singular, uninflected form for adjectives (Juilland et al. 1965, p. XXVII). Plurale tanta (like *scissors*), however, have no singular form, although their stem may occur in compounds (*scissor-movement*). For this and similar reasons we plead for the use of an abstract stem which is not necessarily one of the forms of the paradigm. This notion from linguistics can be used fruitfully in computational lexicology and lexicography.

In a lot of common databases, for example a person registration system, or a library administration system, data-objects correspond to objects in reality, for instance a person, an address, or a book. Identification consists of connecting the object in reality to the object in the data collection. A car can be connected to a licence number. A person can be connected to a social security number.

Lexical objects however, only occur in reality as instantiations of types. A lexicon is not a list of tokens, but one of instantiations that are being realized in specific discourse contexts as tokens of that type. Some of the types in our Lexical Information System (morphemes and stems) are abstract in the sense that they are not instantiated as tokens in reality.

(1b) A phonetic representation of this abstract stem is necessary to relate inflectional forms to a stem. For instance the stem in *entry* and *entries* have the same phonetic form although their spelling differs. A phonetic representation can also be helpful for the more practical reason of correcting a misspelling. We could name the lexicon *orthofonic* (Lurquin 1982).

(2) The lexicon does not have a 'flat' structure, but one that consists of three layers, where information is represented with entities as abstract as possible. All inflectional forms point to their stems; information about for instance subcategorisation is the same for all forms in the inflectional paradigm, and therefore represented with the stem of the paradigm. This implies the presence of two layers in the lexical system.

But a stem consist of morphemes, and therefore the existence of a third layer is presumed: one consisting of morphemes. What morphemes carry are formal properties: for instance, the morpheme *mit* changes into *mission* in *permit* and *admit* if a substantive is derived. Allomorphy, being a matter of form, also resides on the level morphemes.

If words are formed with productive morphological processes, their elements occur in the layered structure carrying the information that is specific for the layer in which the elements reside. This information can then be transferred to layers where less abstract entities reside.

2.2 Information Analysis

The *information analysis* establishes the relations between the information about the lexical objects and the objects themselves in the information system by the use of a so-called "conceptual schema" (see appendix 1 for the complete conceptual schema). Such a schema is useful becase it gives insight into the kinds of information and the constraints on the information, and serves as a means of communication between designers, implementers and users of the information system.

In order to give some insight into the representation technique used here, an example excerpted from the complete conceptual schema in appendix 1 will be given and a description of some relevant aspects of the example.

The collective noun for all objects in the lexicon is 'lexical entity'. These lexical entities are represented in the schema (see Fig. 1) by a circle with the name of the objects concerned. A part of the lexical entities is the group 'stem of form', represented by a circle and an arrow to denote the subset property.

Lexical entities have a primary stress. For example *coffee* has as primary stress on the syllable that begins on the first position in the word. 'Stress' is an object that is included in the Lexical Information System, and is therefore as well represented with a circle. Between brackets is the word 'position' that indicates that 'stress' is denoted with a position.

The squares denote predicate roles that represent the relations of one object with another. The relation between the object groups 'lexical entity' and 'stress' can be read in two directions: 'lexical entity' 'has primary' 'stress', and, 'stress' 'is primary stress of' 'lexical entity'.

To these relations constraining rules may apply: a indicates that a relation applies to all objects that are connected to it; e.g. *'all* lexical entities have a primary stress'. This is in contrast with secondary stresses: these do not occur with all lexical entities.

Another constraint is that some relations occur only once with an object. This is the so-called *unicity-constraint*; 'a lexical entity has *only one* primary stress'. This is indicated with an arrow on the side where the relation is described ↔. A stem or a form can have more than one secondary stress.

In Fig. 1 two constraints between relations are present. A lexical entity cannot have a primary and a secondary stress on the same position. This is shown by an encircled cross between the relations concerned ⊙. If the relations apply together for all objects in one group these objects relations are connected with a ⊗; 'the stresses that are primary stress and those that are secondary stress are together all stresses'.

The complete conceptual schema is shown in Appendix 1.

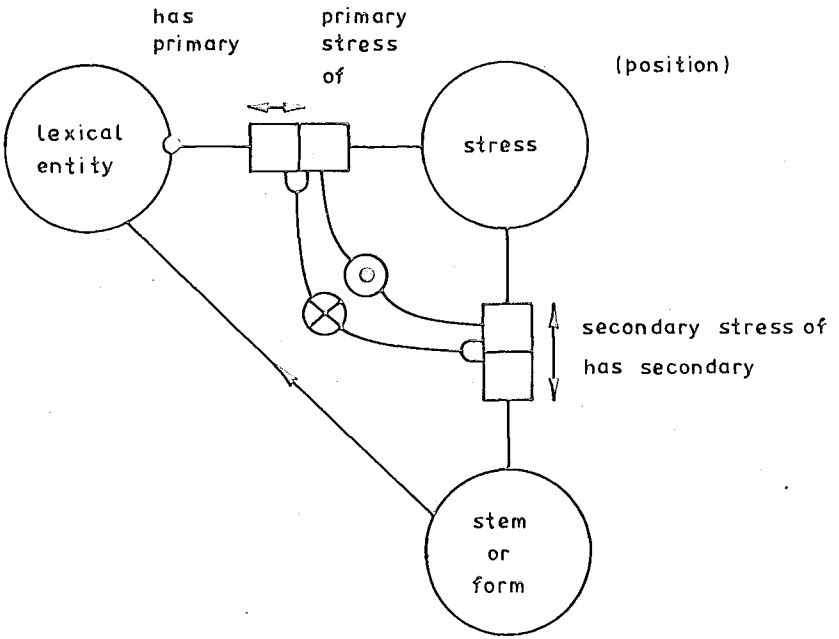


Figure 1

3. Identity

The conceptual schema describes transferable knowledge, information about relates can be described uniquely: only in that case can producer and consumer of information establish the identity of an object in the communication process. This communication process can take place between men, between machines or between man and machine. A computer, however, can only deal with consistently represented information: human users can adequately respond to inconsistencies, computer programs only in a very limited way (Boguraev 1987). Therefore the ability to identify every object uniquely in our Lexical Information System is of great importance.

The unique identification of lexical objects is derived from the identity of the objects. To gain insight into the identity of objects in the lexicon, a list of properties necessary to determine the identity of the objects in the data-collection is constructed during the information analysis besides the conceptual schema[2]. Not all properties of an object are relevant for the determination of identity.

Determining the list of properties requires careful study in cases where objects have more than one value for a property. An example is the word *have* that can function as a noun or as a verb. These are considered non-identical because their categories differ. As a verb, *have* can function as auxiliary or as an non-auxiliary, 'independent' verb. In this case, a choice of one out of several possible representations has to be made. Roughly speaking the choice is between one object with several values for a property (one word *have* that can function as an auxiliary and as an 'independent' verb), and several objects with one value (two words *have*, where one functions as an auxiliary verb, and the other as an 'independent' verb). These choices also mirror the storage of information on the level of the macrostructure or the microstructure, which is common use in lexicography: this distinction equals the division of information in a Lexical Information System between indexing-structure (macrostructure) and information represented within each lemma (microstructure).

No decisive motivation for the choice can be given from a linguistic viewpoint, neither from the perspective of information analysis. Besides that, choices on the conceptual level are not to be determined by realisation aspects such as complexity of the database or ability to conduct the search procedure in the database. The important thing is the interpretation of the linguistic application, and not the possibility of more efficient processing or storage, because this is independent of linguistics. The conceptual representation of the objects may not be influenced by the physical representation.

So, the identity of lexical objects should be determined by linguistic factors, but which factors to use is determined by the objective of the Lexical Information System. Within this objective the lexical objects have a function from which identity-determining properties arise.

Firstly, the objective of the Lexical Information System tells us that the knowledge base has two global characteristics: abstract stems and a layered structure. Secondly, the objective determines how to establish the relation between the information about the objects and the objects themselves. Identity-determining proper-

ties should be 'useful' in this respect in the perception of the users of the application, whether this is a computer-program or a human being. Something is

> "(. . .) perceived and treated as two things, rather than one or three or ninety-eight(..). Not by any natural law, but by the arbitrary decision of some human beings, because the perception was useful to them, and corresponded to the kinds of information they were interested in (. . .)" (Kent 1978: 7).

So, identity cannot be established by claiming that two objects are identical if all properties they posses are equal. The precise distinction between lexical objects can only be made after careful study of the properties: as was noticed already, not all properties are relevant for identity. For words in a language-processing computer system the function of the lexicon with regard to the objective of the application is important: if the objective of the system is only hyphenation, there is no need to distinguish between homonyms. In a system that interprets sentences semantically, a detailed semantic classification is necessary (Van der Linden and De Smedt 1987).

Linguistic factors can be divided into formal, syntactic, structural and semantic factors in our application.

1 — Words are *formally* identical if they have the same written and spoken form. Formal identity is a conditio sine qua non for identification of words. (Schultink 1965: 358)
2 — *Syntactic* differences between formally identical stems are only of interest if defined in terms of valence; as the possibility to combine with other words. c.f. the *have*-case mentioned before.
3 — Utterances that are formally identical, but differ *structurally*, like the classical sentence "Flying planes can be dangerous", are considered different. The objects of linguistic theory are in our view grammatical structures and not their spoken forms (following Higginbotham 1985: 552). Indeed the words have been formed from the same lexical material, but the composition of meaning has taken place along two different paths, and therefore the meaning differs. This argument thus relates to that of semantic identity.
4 — Semantic identity can only exist if a certain correspondence in meaning exists. The problem however, is to determine the "quantity" of agreement. In our lexicon we use a simple *semantic* classification that divides words into classes with respect to a number of properties (Geerts *et al.* 1984) that are chosen because they reveal other linguistic information, and therefore contribute to generalization. For instance mass nouns, like *mud* or *water* are uncountable and therefore don't have plural forms. Therefore words that differ with respect to their semantic classification are considered different.

4. Conclusion

In designing a sophisticated lexical component of a natural language processing system, methods drawn from computer science have shown their usefulness for computational lexicology and lexicography in the formal description of the lexicon. The methods explicate the considerations used in designing the lexicon. The objective of the lexicon is of great importance and determines the structure of the lexicon

in two ways. First, sophisticated applications require detailed lexical information, and therefore a linguistically structured Lexical Information System, with a layered structure and abstract entities. Second, the objective of the lexicon determines in what way linguistic factors divide lexical information between macro- and microstructure.

Notes

[1] Part of the research described in this paper has been carried out as part of ESPRIT-project OS—82: "An Intelligent Multi-Media Office Workstation", work package: "Natural Language Processing.". This part of the research took place at the 'Language Technology Project' at Nijmegen University. An elaborated version of the present paper is Van der Linden et al. 1988.

[2] An elaborate description of this list and the conceptual schema can be found in Van Boven and Van der Linden (1987).

References

Boguraev, B. 1987. 'On-line lexical resources for Natural Language Processing' in *Proceedings of Seminar "Recent developments and applications of natural language understanding"*. London, 8—10 December 1987. 143—160.

Geerts, G., W. Haeseryn, J. de Rooij, and M.C. van den Toorn. 1984. *General Dutch Grammar*. Groningen: Wolters-Noordhof (Dutch).

Higginbotham, J. 1985. 'On semantics' in *Linguistic Inquiry* 16 (4): 547—593.

Juilland, A., P. Edwards, and I. Juilland. 1965. *Frequency Dictionary of Rumanian Words*. Den Haag: Mouton & Co.

Kempen, G., G. Anbeek, P. Desain, L. Konst, and K. De Smedt. 1987. 'Author environments: fifth generation text processors' in DG XIII of the CEC (ed.). *Esprit '86: Results and Achievements*. Amsterdam: North-Holland

Kent, W. 1978. *Data and Reality*. Amsterdam: North-Holland.

Koskenniemi, K. 1983. *Two-level morphology*. Diss. University of Helsinki.

Lurquin, G. 1982. 'The Orthophonic Dictionary' in J. Goetschalckx and L. Rolling (eds.). *Lexicography in the Electronic Age*. Amsterdam: North-Holland Publishing Company.

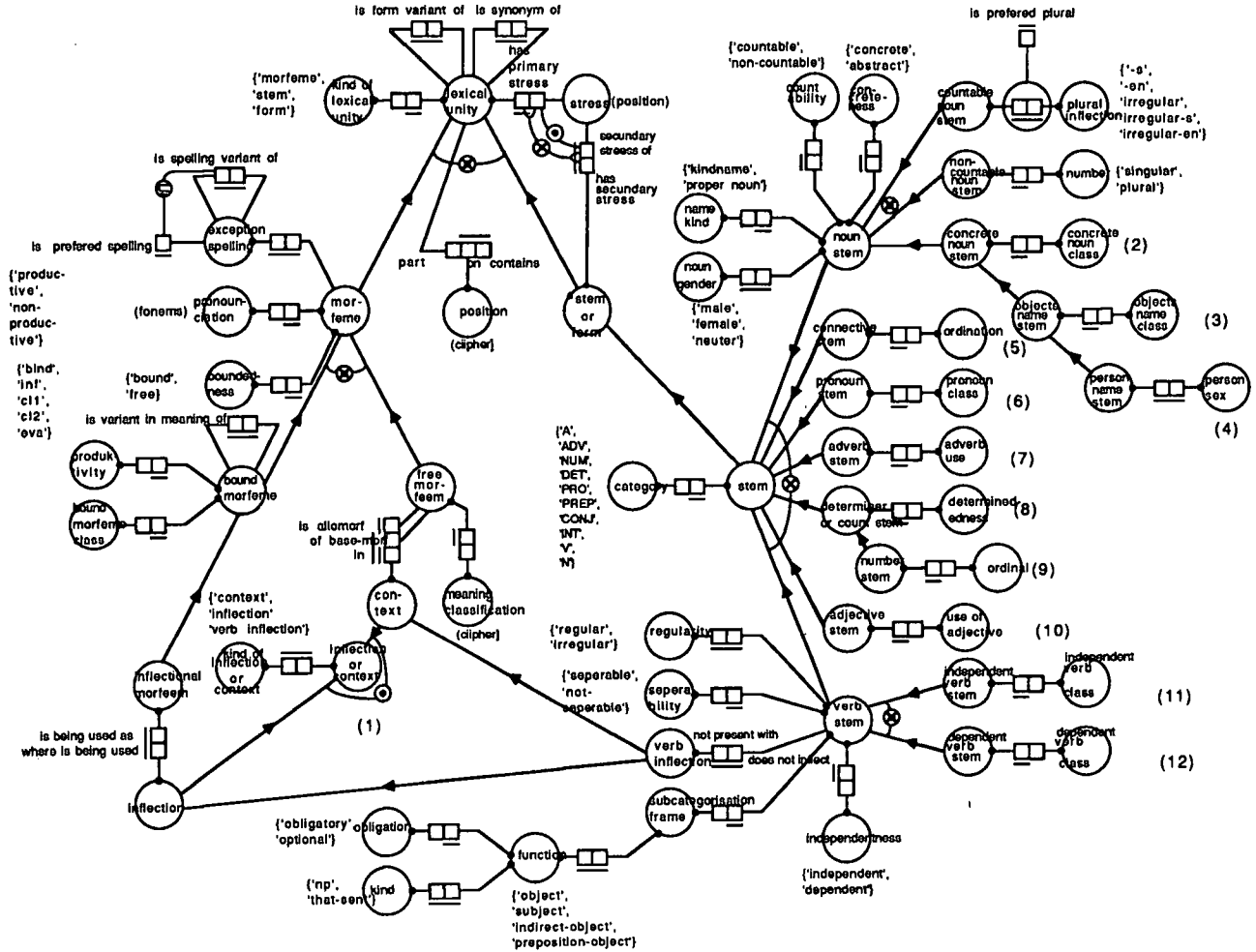
Scalise S. 1984. *Generative Morphology*. Dordrecht: Foris.

Van Boven, P. and M. Van der Linden. 1987. *A lexical information system for applications in language technology*. Masters thesis. Department of Information Systems, University of Nijmegen. (Dutch)

Van der Linden, E. and K. De Smedt. 1987. 'Computer lexicons for an author system' in *Toegepaste taalkunde in artikelen*. 27: 33—41. (Dutch)

Van der Linden, E., P. van Boven, S. Brinkkemper, M. Van der Linden, and K. De Smedt. 1988. *The representation of lexical object*. ITI-TNO internal report No. 88 ITI B21. (Dutch, elaborate version of the present paper)

Verheijen, G., and J. Van Bekkum. 1982. 'NIAM: An Information Analysis Method' in T.W. Olle, H.G. Sol, and A.A. Verrijn Stuart (eds.). *Information System Design Methodologies—A comparative Review*. Amsterdam: North-Holland Publishing Company. 537—589.


is form variant of
is synonym of
has primary stress
{'morfeme', 'stem', 'form'}
kind of lexical unity
lexical unity
stress (position)
secundary stress of
has secundary stress
is spelling variant of
exception spelling
is prefered spelling
{'productive', 'non-productive'}
(fonems)
pronunciation
morfeme
part
on contains
position
(cipher)
stem or base
{'bind', 'inf', 'cl1', 'cl2', 'eva'}
{'bound', 'free}
boundedness
is variant in meaning of
produktivity
bound morfeme
bound morfeme class
free morfeem
is allomart of base-morf in
context
meaning classification
(cipher)
{'context', 'inflection' 'verb inflection'}
kind of inflection or context
inflection or context
(1)
inflectional morfeem
is being used as where is being used
inflection
{'regular', 'irregular'}
regularity
{'seperable', 'not-seperable'}
seperability
verb inflection
not present with
does not inflect
subcategorisation frame
{'obligatory' 'optional'}
obligation
function
{'np', 'that-sent'}
kind
{'object', 'subject', 'indirect-object', 'preposition-object'}
{'A', 'ADV', 'NUM', 'DET', 'PRO', 'PREP', 'CONJ', 'INT', 'V', 'N'}
category
stem
is prefered plural
{'countable', 'non-countable'}
{'concrete', 'abstract'}
countability
concreteness
countable noun stem
plural inflection
{'-s', '-en', 'irregular', 'irregular-s', 'irregular-en'}
{'kindname', 'proper noun'}
name kind
noun gender
{'male', 'female', 'neuter'}
noun stem
non-countable noun stem
number
{'singular', 'plural'}
concrete noun stem
concrete noun class
(2)
objects name stem
object name class
(3)
connective stem
ordination
(5)
person name stem
person sex
(4)
pronoun stem
pronoun class
(6)
adverb stem
adverb use
(7)
determiner or count stem
determinedness
(8)
number stem
ordinal
(9)
adjective stem
use of adjective
(10)
verb stem
independent verb stem
independent verb class
(11)
dependent verb stem
dependent verb class
(12)
independentness
{'independent', 'dependent'}

(1) inflection or context code:

{'n-mv', 'a-ver',
'a-gro', 'a-ove',
'a-dat', 'a-gen',
'v-inf', 'v-te1', 'v-te2', 'v-te3',
'v-tmv', 'v-pen', 'v-pmv', 'v-tde', 'v-vde',
'v-aen', 'v-ien', 'v-imv',
'w-com', 'w-der', 'n-dim'}

(2) concrete noun class code:

{'mass noun',
'collective noun',
'object name noun'}

(3) object name class code:

{'persons name',
'animal name',
'thing name'}

(4) person sex code:

{'male',
'female'}

(5) ordination code:

{'sub-ordinating',
'co-ordinating'}

(6) pronoun code:

{'personal',
'demonstrative',
'possessive',
'relative',
'wh',
'reflexive',
'undetermined',
'exclamative'}

(7) adverb usage code:

{'adjectival-and-adverbial',
'only adverbial'}

(8) determinedness code:

{'determined',
'not determined'}

(9) number code:

{'ordinal number',
'cardinal number'}

(10) adjective use code:

{'attributive',
'non-attributive',
'attributive and non-attributive')

(11) independent verb class code:

{'transitive',
'intransitive',
'transitive and intransitive',
'reflexive',
'reflexive and intransitive'}

(12) dependent verb class code:

{'auxiliary',
'copula',
'impersonal'}