Towards a Formal Representation of Components of German Compounds

Thierry Declerck
DFKI GmbH
D-66123 Saarbrücken, Germany
&
Austrian Centre for Digital Humanities
A-1010 Vienna, Austria
declerck@dfki.de

Piroska Lendvai
Dept. of Computational Linguistics
Saarland University
D-66123 Saarbrücken, Germany
piroska.r@gmail.com

Abstract

This paper presents an approach for the formal representation of components in German compounds. We assume that such a formal representation will support the segmentation and analysis of unseen compounds that feature components already seen in other compounds. An extensive language resource that explicitly codes components of compounds is GermaNet, a lexical semantic network for German. We summarize the GermaNet approach to the description of compounds, discussing some of its shortcomings. Our proposed extension of this representation builds on the *lemon* lexicon model for ontologies, established by the W3C Ontology Lexicon Community Group.

1 Introduction

The motivation for our study is the assumption that the availability of a formal representation of components in German compound words can help in the detection, processing and analysis of new unseen compounds. Compounding in German is a productive process to create (new) words, and these typically consist of lexical items partly seen in other compounds already. Our aim is to create a formal representation of such components in German, in order to facilitate the segmentation and possibly the generation of compound words.

Several German lexical resources feature designated elements to mark up entries as compounds, whereas they typically lack elements that would represent the components of compounds.

One of the most fully-fledged resource of German compound nouns is GermaNet (Hamp and Feldweg, 1997; Kunze and Lemnitzer, 2002), a lexical semantic net for German following the principles of the Princeton WordNet (Fellbaum, 1998). The approach of GermaNet to the encoding of elements of compounds within the lexical-semantic net is described in (Henrich and Hinrichs, 2011). Additionally to this representation of compounds, GermaNet offers a freely available list of 66,047 nominal compounds split to their modifier(s) and head, in a tab-delimited (tsv) format[1]. Table 1 shows few examples taken from this list, while Example **??** shows the encoding of the compound *Rotsperre* ('red card suspension') within the XML representation of the GermaNet lexical semantic network.

Based on the GermaNet description of compounds, several studies on annotating and sys-

[1] http://www.sfs.uni-tuebingen.de/GermaNet/documents/compounds/split_compounds_from_GermaNet11.0.txt

tems on processing compounds have been proposed (Hinrichs et al., 2013; Santos, 2014; Dima et al., 2014; Dima and Hinrichs, 2015).

Compound	Modifier(s)	Head
Rotschopf	rot	Schopf
Rotschwanz	rot	Schwanz
Rotschwingel	rot	Schwingel
Rotspecht	rot	Specht
Rotsperre	rot	Sperre
Rotstich	rot|Rot	Stich
Rotstift	rot	Stift
Rotstiftaktion	Rotstift	Aktion

Table 1: Examples from the GermaNet list of nominal compounds.

The few examples listed in Table 1 show that GermaNet describes explicitly only immediate constituents of compounds, but is also reflecting the recursive nature of compounds that have more than two constituent parts, as can be seen with the words *Rotstift* ('red pencil') and *Rotstiftaktion* ('cutback', 'reduce spending'). In this case a tool can easily split *Rotstiftaktion* into *rot*, *Stift* and *Aktion*, on the basis of the segmentation of *Rotstift*.

We note also that one compound can have more than one modifier, as in the case of *Rotstich* ('tinge of red'), where we have both an adjectival (*rot*) and a nominal (*Rot*) modifier. GermaNet marks the different part-of-speech (PoS) properties of the components being in the modifier position by using different cases: Upper case marks a noun (as this is the case for all the listed compounds), while lower case marks either a verb or an adjective.

We observe also that the modifier *rot* is often repeated (in fact much more often then in this slice taken from the list: there are also many compounds ending with the component *rot*).

In the following sections we present first the GermaNet formal representation of compounds in the full context of the lexical semantic net. Then we suggest our extensions to the GermaNet representation, utilizing modules of the *lemon*[2] approach to the encoding of lexical data.

2 Representation of Compounds in the GermaNet lexical semantic net

The structure of a GermaNet entry containing the compound word *Rotsperre* ('red card suspension') is shown in Example 1. The relevant information is to be found in the XML elements rendered in bold face.

```
<synset class="Geschehen"
    category="nomen" id="s21159">
  <lexUnit id="l29103"
      styleMarking="no"
      artificial="no"
      namedEntity="no" source="
      core" sense="1">
    <orthForm>Rotsperre</
        orthForm>
    <compound>
      <modifier category="
          Adjektiv">rot</
          modifier>
      <head>Sperre</head>
    </compound>
  </lexUnit>
  <paraphrase>beim Fussball</
      paraphrase>
</synset>
```

Example 1: A compound lexical unit in GermaNet: *Rotsperre* ('red card suspension')

In this formal representation, the PoS of the modifier element of the compound (*rot*, 'red') is explicitly given, while this is not the case for the head, as the PoS of the head element of a compound is identical to the PoS of the whole compound. However, we advocate that explicitly encoding the PoS information of the

[2]The *lexicon model for ontologies* (lemon) is resulting from the work of the W3C Ontology Lexicon Community Group; https://www.w3.org/community/ontolex/wiki/Final_Model_Specification.

head component can be necessary; for example if a tool would access only the repository of components. In this case, the tool would have to infer the PoS information of the head component from the compounds in which it occurs, adding thus an additional processing step, which can be avoided if the PoS of the head component is explicitly marked.

As already observed for the list of compounds in the tsv format, the GermaNet entry displays here the adjective modifier in lowercase. In this case we are loosing the information about the original use of the word form. We suggest to introduce an additional feature in which the original form of the component is preserved.

By observing the list of compounds provided by GermaNet, we noted that the modifier component of *Rotsperre* keep recurring in other compounds. This is for sure also the case for the head components. For example, the component *Sperre* ('suspension', 'block', ...) is repeated in the related word *Gelbsperre* ('yellow card suspension'). Such productively recurring components would be beneficial to have encoded in a repository so that they are included only once in a lexicon, possibly with links to the different components they can be combined with, depending on their related senses.

The use of a modifier in a compound can play a disambiguation role. While we can easily establish a relation between the reduced set of senses of the compound and the set of senses of the head of the compound, we have no immediate information on the synsets associated to the modifier of the compound. This is an information we would also like to explicitly encode.

Further, we consider the encoding of the *Fugenelement* ('connecting element') that is often used in the building of compounds; e.g. the *s* in *Führungstor* ('goal which gives the lead'). GermaNet does not include this information in its XML representation.

Finally, we notice that the ordering of components is not explicitly encoded.

In order to remedy the above issues, we suggest to adopt the recently published specifications of the *lemon* model. In the following section, we describe this model and our suggested representation of GermaNet compounds.

3 The *lemon* Model

The *lemon* model has been designed using the Semantic Web formal representation languages OWL, RDFS and RDF[3]. It also makes use of the SKOS vocabulary[4]. *lemon* is based on the ISO Lexical Markup Framework (LMF)[5] and the W3C Ontology Lexicon Community Group proposed an extension of the original *lemon* model[6], stressing its modular design.

The core module of *lemon*, called *ontolex*, is displayed in Figure 1. In *ontolex*, each element of a lexicon entry is described independently, while typed relation markers, in the form of OWL, RDF or *ontolex* properties, are interlinking these elements.

Additionally to the core module of *lemon*, we make use of its decomposition module, called *decomp*[7], designed for the representation of Multiword Expression lexical entries, and which we use for the representation of compound words.

[3] See respectively `http://www.w3.org/TR/owl-semantics/`, `https://www.w3.org/TR/rdf-schema/`, and `https://www.w3.org/RDF/`

[4] `https://www.w3.org/2004/02/skos/`

[5] See (Francopoulo et al., 2006) and `http://www.lexicalmarkupframework.org/`

[6] See (McCrae et al., 2012)

[7] `http://www.w3.org/community/ontolex/wiki/Final_Model_Specification`

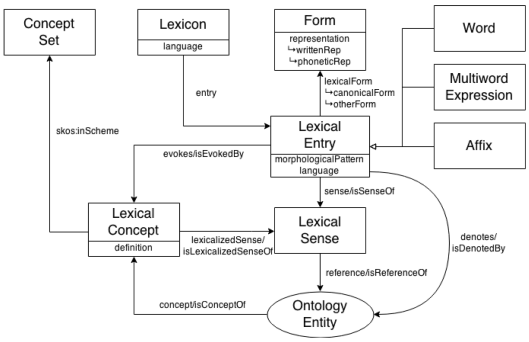

Figure 1: *ontolex*, the core module of *lemon*. Figure created by John P. McCrae for the W3C Ontolex Community Group.

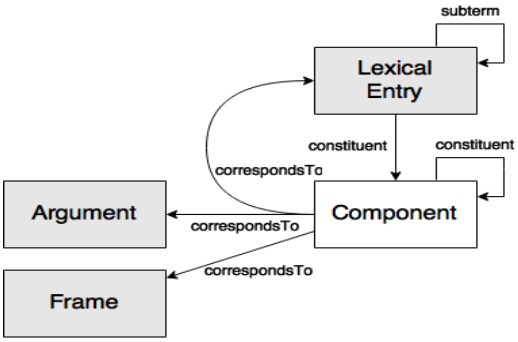

Figure 2: *decomp*, the decomposition module of *lemon*. Figure created by John P. McCrae for the W3C Ontolex Community Group.

The relation of *decomp* to the core module, and more particularly to the class ontolex:LexicalEntry, is displayed in Figure 2. Components of a compound (or a multiword) entry are pointed to by the property: decomp:constituent. The range of this property is an instance of the class decomp:Component.

Taking again *Rotsperre* ('red card suspension') as an example, and which is built of two components, we make use two times of the decomp:constituent property, the current values of it being `:Rot_comp` and `:sperre_comp` (see the corresponding RDF code given below in the entries (1-3), which are instances of the class `ontolex:Component`. This way we can encode the surface forms of the components, as they are used in compounds.

The relation between the `ontolex:Component` instances (the surface forms of the components occurring in the compounds) and the `ontolex:Word` instances (the full lexical entries corresponding the surface form of the components) follows the schema for the relation between the two classes `ontolex:LexicalEntry` and `ontolex:Component`, which is graphically shown in Figure 2. For our example *Rotsperre*, as shown in the entries (1-3) below, the elements *Rot* and *sperre* are instances of the class `ontolex:Component`, and as such *sperre* can be linked/related to other compounds like *Löschsperre* ('deletion block') or to the (semantically more closely related) *Gelbsperre* ('yellow card suspension'). The property `decomp:correspondsTo` links the components to the lexical entries that encode all the lexical properties of those surface forms used in the compound word.

A simplified representation of the compound entry *Rotsperre* and of its components is displayed below, in (1-3). In the entry (1) we use `rdf_1` and `rdf_2`[8] for marking the order of the two components in the compound word. We assume that information on the position of the elements can be relevant for the interpretation of the compound.

(1) :Rotsperre_lex
rdf:type ontolex:LexicalEntry ;
lexinfo:partOfSpeech lexinfo:noun ;
rdf:_1 :Rot_comp ;
rdf:_2 :sperre_comp ;

[8] As instances of the property `rdfs:ContainerMembershipProperty`, see `http://www.w3.org/TR/rdf-schema/` for more details.

decomp:constituent :Rot_comp ;
decomp:constituent :sperre_comp ;
decomp:subterm :Sperre_lex ;
decomp:subterm :rot_lex ;
ontolex:denotes
<https://www.wikidata.org/wiki/
Q1827> .

Entries (2) and (3) below show the encoding of the instances of the class `decomp:Component`:

(2) :Rot_comp
rdf:type decomp:Component ;
decomp:correspondsTo :rot_lex .

(3) :sperre_comp
rdf:type decomp:Component ;
decomp:correspondsTo
:Sperre_lex .

The proposed approach to the representation of elements of compounds seems intuitive and economical, since one component can be linked to a large number of other components, and, next to decomposition, can also be used for the generation of compound words, taking into account the typical position such components are taking in known compounds.

In the compound entry (1) we also make use of the property `decomp:subterm`. This property links the compound to the full lexical information associated to its components, including the senses of such components. The motivation of the *lemon* model is the determination of senses of lexical entries by reference to ontological entities outside of the lexicon proper. We can thus easily extend the representation of the compound word with sense information, by linking the components and the compound word to relevant resources in the Linked Open Data (LOD) cloud. The sense of `:Rot_comp` is given by a reference to `http://de.dbpedia.org/page/Rot`, where additional associations of *red* with political parties or sports clubs, etc. can be found. The same holds for `:sperre_comp`, which can be linked to the LOD resource `http://de.dbpedia.org/page/Sperre`.

Additionally, for the sense of the complete compound word we link to the LOD resource: `https://www.wikidata.org/wiki/Q1827`, with the specific meaning of suspension from a sports game. The senses repository for *Sperre* can look as displayed in the lexicalSense entries (4) and (5).

(4) :sperre_sense1
rdf:type ontolex:LexicalSense ;
rdfs:label "A sense for the German word 'Sperre"' @en ;
ontolex:isSenseOf :Sperre_lex ;
ontolex:reference
<http://de.dbpedia.org/resource/Lock>
.

(5) :sperre_sense2
rdf:type ontolex:LexicalSense ;
rdfs:label "A sense for the German word 'Sperre"' @en ;
ontolex:isSenseOf :Sperre_lex ;
ontolex:reference
<http://de.dbpedia.org/resource/
Wettkampfsperre> .

Our current work includes associating GermaNet senses as values of the `ontolex:LexicalSense` property. We are also encoding connecting elements (*Fugenelemente* with the help of the `ontolex:Affix` class.

4 Conclusion

We presented an approach for the formal representation of elements that occur in compound words. Our motivation is to provide rules for computing compound words on the basis of their components.

Acknowledgments

Work presented in this paper has been supported by the PHEME FP7 project (grant No. 611233) and by the FREME H2020 project (grant No. 644771). The author would like to thank the anonymous reviewers for their very helpful comments.

References

Corina Dima and Erhard Hinrichs. 2015. Automatic noun compound interpretation using deep neural networks and word embeddings. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 173–183, London, UK, April. Association for Computational Linguistics.

Corina Dima, Verena Henrich, Erhard Hinrichs, and Christina Hoppermann. 2014. How to tell a schneemann from a milchmann: An annotation scheme for compound-internal relations. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Gil Francopoulo, Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Y Pet, and Claudia Soria. 2006. Lexical markup framework (lmf. In *In Proceedings of LREC2006*.

Birgit Hamp and Helmut Feldweg. 1997. Germanet - a lexical-semantic net for german. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Verena Henrich and Erhard W. Hinrichs. 2011. Determining immediate constituents of compounds in germanet. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *RANLP*, pages 420–426. RANLP 2011 Organising Committee.

Erhard Hinrichs, Verena Henrich, and Reinhild Barkey. 2013. Using partwhole relations for automatic deduction of compound-internal relations in germanet. *Language Resources and Evaluation*, 47(3):839–858.

Claudia Kunze and Lothar Lemnitzer. 2002. Germanet - representation, visualization, application. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Canary Islands - Spain, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L02-1073.

John P. McCrae, Guadalupe Aguado de Cea, Paul Buitelaar, Philipp Cimiano, Thierry Declerck, Asunción Gómez-Pérez, Jorge Gracia, Laura Hollink, Elena Montiel-Ponsoda, Dennis Spohr, and Tobias Wunner. 2012. Interchanging lexical resources on the semantic web. *Language Resources and Evaluation*, 46(4):701–719.

Pedro Bispo Santos. 2014. Using compound lists for german decompounding in a back-off scenario. In *Workshop on Computational, Cognitive, and Linguistic Approaches to the Analysis of Complex Words and Collocations (CCLCC 2014)*, pages 51–55.