Syntactic Analysis of Phrasal Compounds in Corpora: a Challenge for NLP Tools

Carola Trips

Universität Mannheim
L13, 9, 68131 Mannheim
ctrips@mail.uni-mannheim.de

Abstract

The paper introduces a "train once, use many" approach for the syntactic analysis of phrasal compounds (PC) of the type XP+N like *"Would you like to sit on my knee?" nonsense*. PCs are a challenge for NLP tools since they require the identification of a syntactic phrase within a morphological complex. We propose a method which uses a state-of-the-art dependency parser not only to analyse sentences (the environment of PCs) but also to compound the non-head of PCs in a well-defined particular condition which is the analysis of the non-head spanning from the left boundary (mostly marked by a determiner) to the nominal head of the PC. This method contains the following steps: (a) the use an English state-of-the-art dependency parser with data comprising sentences with PCs from the *British National Corpus* (BNC), (b) the detection of parsing errors of PCs, (c) the separate treatment of the non-head structure using the same model, and (d) the attachment of the non-head to the compound head. The evaluation of the method showed that the accuracy of 76% could be improved by adding a step in the PC compounder module which specified user-defined contexts being sensitive to the part of speech of the non-head parts and by using *TreeTagger*, in line with our approach.

Keywords: Phrasal compounds, state-of-the-art dependency parser, compounder

1. State of the art

This paper problematizes the syntactic analysis of phrasal compounds extracted from English corpora and proposes a "train-once, use-many approach" which uses a state-of-the art dependency parser for the analysis of full sentences and as a compounder for the non-head of phrasal compounds.

In many languages compounding is a productive process of word formation. This is the case for example in German and English where different types of compounds exist with Noun+Noun compounds (NNC) being most frequent. In these languages the structure of NNCs is such that the formal and semantic head is right-headed with the determining or modifying non-head preceding it (e.g. German *Haustür* 'door of a house', English *house guest*). Although both languages exhibit NNCs, unlike English, German does not graphically split the non-head and the head, which results in ambiguities in automatic analysis. Thus, the task of finding and segmenting unknown compounds in text corpora is at stake here, and a number of methods (for different purposes) has been proposed. For example, Brown (2002) dealing with parallel English-German texts in the medical domain, proposes a method for splitting compound words into their constituents based on cognate words in the other language of a parallel corpus. Schiller (2005) discusses an experiment where she used a finite-state morphological analyzer to segment compounds which is based on weights for compound segments. Daðason & Bjarnadóttir 2014 describe a method where a decompounder which estimates the probability of a constituent in an Icelandic unknown compound on the basis of known compounds splits compounds into binary constituents. In all of these cases NLP tools serve to analyse and segment NNCs.

A major motivation for the approach described here is that many of the current approaches deal with the more predictable types of NNCs. This is especially true for lexicon-based or ontology-based approaches which hold the view that 'many nominal compounds are fixed expressions' (McShane et al., 2014, 1; the authors further state that this 'has been long-recognized by linguists'). Although it is certainly true that a number of NNCs in languages showing this type of word formation are fixed entities, a fair amount of them is nevertheless subject to a productive word formation rule. One of the most convincing examples here is Icelandic with its many ways to productively build NNCs, but clearly also German and English. Thus, the task is not only to identify non-transparent NNCs (which are actually less interesting from a linguistic point of view) but transparent ones built on the fly by productive rules. Our approach is supported by other NLP-oriented studies that acknowledge not only the high proportion of words which are part of NNCs in average text (up to 3.9% in Reuters, Nakov, 2013) but also their high degree of productivity requiring that they 'be interpreted compositionally' (*ibid*).

It may be true that analyses based on ontological relations or more or less fixed lexicon entries work well for particular, often domain-specific text types, but we are convinced that the automatic processing of 'normal' text types requires a more flexible approach. This is particularly true of languages having right-headed compounds, as for example the Germanic languages, where the non-head can expand in a rather unconstrained way and even contain phrases like VPs or even full sentences. These compounds are therefore called phrasal compounds (PCs). This contrasts with the very limited structural productivity of languages with left-headed NNCs (cf. the volume of Trips and Kornfilt, 2015 for a contrastive account and Haider, 2013 for a theoretical account of these differences).

The ideal method for coping with productive but unpredictable structures as we find them in PCs would be the use of a standard tool in a well-defined, particular condition.

This particular condition can be defined as the part of the compound spanning from the left boundary (mostly marked by a determiner) up to the head of the compound when the morphological complex is parsed. Such a method is attractive from the point of view of linguists (i.e. their limited desire or ability to use highly specialised linguistic tools), but also from a cognitive point of view, since it can be seen as a simulation of the situation of the hearer/reader who, at a given point in hearing/reading has to interrupt the parsing process of the sentence and analyse the phrasal non-head of such a complex compound. We try to simulate this process by using a "train-once, use-many" approach: (a) we use an English state-of-the-art dependency parser with data comprising sentences with PCs from the *British National Corpus* (BNC); (b) we identify where and how the parse of complex compounds fails, (c) we use a PC compounder (perl script) which (i) splits the non-head and the head and analyses both separately (ii) adds user-defined contexts to correctly analyse the non-head, (iii) attaches the correctly analysed non-head to the compound head.

2. Phrasal Compounds

As noted above, phrasal compounds are a special type of compound in that it includes a syntactic complex in a morphological structure. PCs have challenged linguistic theories based on the *Lexical Integrity Hypothesis* which postulates a strict demarcation between morphology and syntax, and—as stated above—they also seem to challenge NLP tools like state-of-the-art parsers.

Phrasal compounds have the following properties (cf. Meibauer, 2003, Trips, 2016):

(i) they are right-headed
(ii) they have only nominal heads
(iii) they have the structure YP + X where YP semantically determines the head (determinative compounds)
(iv) they exhibit a phrasal intonation pattern of the non-heads
(v) they may show anaphorical binding into phrasal non-heads

The data basis of this paper is a corpus-study of English PCs in the BNC. It revealed that they predominantly occur in writing and the phrasal non-head is either marked by quotation marks, or the elements of the phrasal non-head are linked by hyphens (cf. Trips, 2012, 2016). Two examples are given in (1):

(1) a. Instead, the exhibits may (for example) be presented in a *"gee-whiz, would you believe it?" fashion* (B7G 1264).
b. Sometimes it's just force of habit – the *it's four-o'clock-so-I-must-want-a-biscuit syndrome* (G36 1947).

Their complex structure has consequences for extracting them from corpora. The data set used here was extracted by querying the *BNCweb* based on strings and POS-tags. It comprises 4715 tokens for PCs with hyphens, and 1926 PCs with quotation marks (the numbers refer to manually cleaned data; note that the hyphenated non-heads do not pose problems to tokenization). From a formal and semantic perspective a differentiation between PCs containing a predicate and those not containing a predicate has proven to be relevant for linguistic analysis. For our method proposed below, however, it does not figure. The crucial relation is the modifier-head relation which applies to all PCs.

3. The grammar of Phrasal Compounds

Coming back to the studies briefly discussed above, a parallel can be drawn: despite the fact that NNCs morphologically differ from PCs, they are still subject to the same dependency relation: both consist of a head which is determined by a preceding non-head (modifier-head relation). But whereas the task of NLP tools for NNCs in languages like German or Icelandic (where NNCs form a graphical unit) is to segment and identify the two nouns building an NNC (i.e. a decompounder), the task of an NLP tool analysing PCs in languages like English and German is to compound a nominal head with its directly preceding non-head in the shape of a phrase, maximally a CP. In addition, due to the semantic nature of PCs they almost exclusively occur with either an indefinite or definite determiner which precedes the non-head in English (and German). Thus, we have two types of indicators of PCs: the first type is a determiner marking the beginning of the PC followed by a quotation mark enclosing the non-head, followed by the nominal head of the PC. In the second case all the elements of the non-head are linked by hyphens. These indicators can be used to identify the succession of words which require a special analysis. This analysis should be able to detect the relations which hold between the words of the non-head and attach the whole structure under the head. In analogy to the 'decompounder' suggested for languages like Icelandic or German we call this tool a PC-compounder. According to what we said in the introduction, this compounder is not a tool in its own right. Instead we propose to use the same general-purpose parser that also provides us with the syntactic structures of the sentences.

4. Method

The problem arises when a parser encounters a PC with quotation marks in a text. We tested this by analysing the output file of a search for PCs in the BNC with a state-of-the-art dependency parser. The choice of the parser is not of particular relevance. We experimented with some of the *mate* tools parsers, e.g. Björkelund et al. (2010) and Bohnet and Nivre (2012). Structures quoted in this paper were produced using the 2010 graph parser with the English parser and tagger models available on the *mate* tools website[1]. One of these erroneous structures is given in Figure 2.

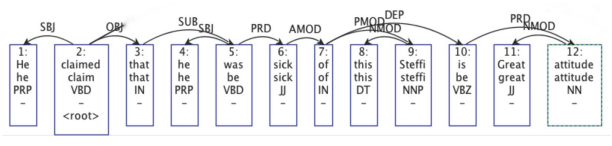

Figure 1: Example of the erroneous analysis of a PC

[1] http://code.google.com/p/mate-tools. We used version anna-3.3.jar.

The non-head of the PC is the copula sentence *"Steffi is Great"* (written in upper case, for a comment see section 5.). The subject of the sentence (*Steffi*) is incorrectly analysed as the nominal head of an NP *this Steffi*. Further, the adjectival complement *Great* is incorrectly analysed as a prenominal modifier of *attitude*, the head noun of the PC (regardless of the quotation marks). A correct parse of this type of PC would analyse the finite verb of the non-head and the determiner preceding it as modifiers depending on the head.

The method we propose to remedy these parsing errors is to use a compounder. The shell script pc-compounder.sh runs the complete annotation procedure. The script requires the Perl script pc-compounder.pl and the mate tools parser. The steps of the procedure are:

1. conversion of the input text to CoNLL. In this step, the Perl script identifies the first pair of brackets in each input line as markers for the non-head constituents of the PC, and the following word as the head of the PC. The script uses the 'feature' column to mark the parts of the PC: pcl=leftmost part, pcm=middle parts (=non-head of PC), pcr=rightmost part (=head of the PC). It creates two temporary CoNLL files[2]:

   i) tmp-pc-input.conll is the converted input text, with non-head PC parts grouped into a single 'word', e.g. *the "hit and run" strategy* to *hit–and–run strategy*. This is to avoid that the erroneous analysis of the non-head affects the analysis of the whole sentence.

   ii) tmp-pc-nonheads.conll contains the non-head parts of the PC embedded in a fake context. This is to enable the parser to analyse its internal structure as correctly as possible.

2. Each of the two CoNLL files generated in step 1 are parsed, using the freely available models for English distributed for the mate tools parser. The output files are named *.dep.conll. Since we favour an approach which does not require an especially trained 'word structure parser', the same models are used for tmp-pc-nonheads.conll and the surrounding context for tmp-pc-input.conll[3].

3. The Perl script joins the two parsed output files by integrating the parsed non-head elements (from tmp-pc-nonheads.dep.conll) into their original position, thus replacing the grouped form in tmp-pc-input.dep.conll. The non-head forms are surrounded by quotes. The PC markers in the 'feature' column are preserved to simplify the verification of the PC analysis[4]. The script has to adapt the 'head' columns to create the correct attachments of the inserted parts, in particular:

   - for the inserted PC parts: adapt their 'head' to the new position within the sentence
   - for the highest node of the PC parts: attach the node to the PC head with the label NMOD
   - for words following the PC: adapt 'head' columns

   The output file is pc-output.conll.

5. Evaluation

The method proposed here was evaluated by checking the parses of the first 500 PCs from a total of 1926 PCs (originally marked with quotation marks) in the file pc.output.conll by using the *What's Wrong With My NLP* tool. Four aspects were examined: (i) the correct attachment of the non-head with the head, (ii) the correct tags of the words within the non-head, (iii) the correct dependencies within the non-head, (iv) the correct label of dependencies within the non-head. The number and rate of the correct analyses are given in Table 1.

Attachment within non-head			Attachment of non-head
Tag	Attach	Label	
349 (70%)	390 (78%)	400 (80%)	382 (76%)

Table 1: Evaluation rates of proposed method

First, the best results are gained with non-heads that are full sentences without ellipsis. Here, all the aspects defined above are correct. Some examples are *"Freud reduces everything to sex" order*, *"large is beautiful" policy*, *"I'm so fat I could be a Turkey" type of robin*, *"Would you like to sit on my knee?" nonsense* or *"Steffi is Great" attitude* (see Fig. 2).

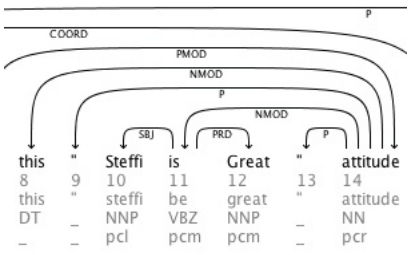

Figure 2: Correct parse of full sentence non-head

Second, a striking result is that only 70% of the PCs are correctly tagged, i.e. all elements part of the non-head and head are analysed correctly in terms of word category. More crucially, an incorrect analysis of word category may result in incorrect parses. On closer inspection, most of the tagging errors are due to spelling: very often elements of the non-head are spelt with a capital, e.g. *"Statement of Case" look*, *"Sponsor A Pig" scheme*, *"So Many Shopping Days to Christmas" idea*, or *"Bring and buy" coffee*. In all of these cases the words spelt with a capital letter are analysed as proper nouns (NNP). This error does not bear on nouns but it does on verbs if they occur on the left edge of the non-head phrase. In the example *"Say No To Strangers"*

[2]The command to run the script is pc-compounder.sh -i <input file> -s.

[3]The command to run the script is pc-compounder.sh -p.

[4]The command to run the script is pc-compounder.sh -j.

campaign, *say* is not analysed as a verb but as a noun and, consequently, the structure of the phrase in a graph is not correct (see Fig. 3).

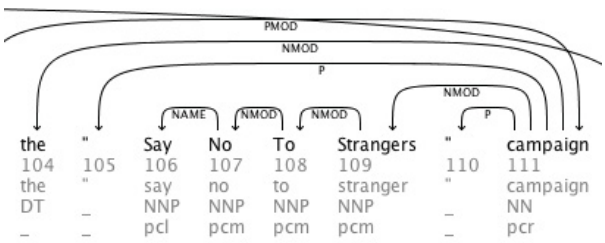

Figure 3: Incorrect parse due to spelling

As the graph shows, the verb *say* is not attached to the head *campaign* with the label NMOD, *no* is not analysed as object of the verb, *to* is not analysed as an adverbial of the verb, and *strangers* is not analysed as PMOD of *to*. The same applies to examples like *"Knit a pattern" menu* where *knit* is analysed as noun and thus not identified as the node that stands in a NMOD relation with the head *menu*. Further, the relation between *pattern* and *knit* is not analysed as an OBJ relation. Other examples of this type are *"Name the Doll" competition* and *"Hang the blessed DJ" lyric*. If the verb occurs with small capitals as in *"slow it down" mode* or *"made in Japan" tag* the tags as well as the parse are correct (see Fig. 4).

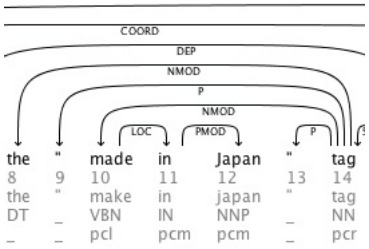

Figure 4: Correct parse of PC with verb + adjunct as non-head

However, this 'rule' is not reliable, i.e. there are cases where verbs in the non-head are spelt in lower case and yet are not correctly analysed (incorrect tag, incorrect parse). Some further examples of this type which display parsing errors are *"poke and dig" method*, *"kick me please" type fashion*, or *"show the shirt routine"* (see Fig. 5). It seems that the application of the rule 'identify an element as (proper) noun occurring after a determiner' is much more probable than the application of the rule 'identify an element preceding a noun as a verb'.

Other problems arise, not suprisingly, with phrases that are highly elliptical. These can be sentences where the head noun of an NP functioning as subject is elided as *"old sounds best" theory* (based on something like *old songs sound best*), where the subject is totally elided as in *"hear no evil, see no evil" brigade* (based on something like *I/they hear no evil, I/they see no evil*) or where verbs and other material are elided as in *"up yours, Vivien, fuck you Thatcher" explosion*[5]. In these cases the parser is at a loss (see Fig. 6).

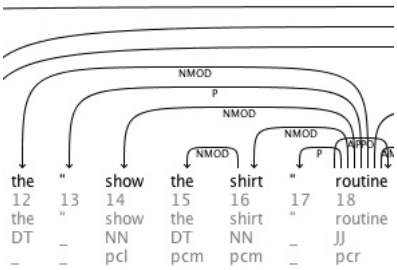

Figure 5: Incorrect parse despite correct spelling

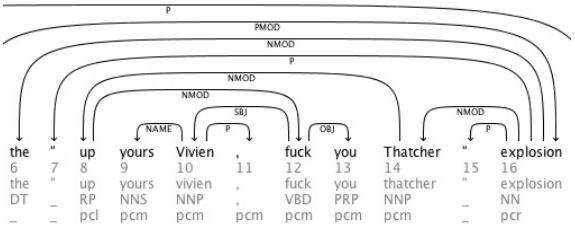

Figure 6: Incorrect parse due to ellipsis

The overall results of the evaluation process are that (i) the error rate of the predicted tags is too high and that (ii) more contexts for the non-heads are needed that include full verb phrases and elliptical structures. To remedy these errors and thus improve our results, we decided to refine our procedure in the following way: the file tmp-pc-non-heads.conll is improved by specifying user-defined contexts which are sensitive to the part of speech tags of the non-head parts. These contexts must be defined in the file pc-contexts.txt.

Following our principle of using only available and published resources, to improve the tagging rate we use *TreeTagger* (Schmid:1997a) with English parameters available on the *TreeTagger* web site[6]. The compounder shell script analyses the input file pc-contexts.txt[7]. The perl script reads the tagger output and matches it with the tag string specified for each user-defined context. This set of rules is given

[5]The sentence in which this PC occurs is: *He seemed somewhat seperate from the "up yours, Vivien, fuck you Thatcher" explosion*. This piece of information does not really help to identify the verb and the noun that are missing in the first part of the sentence. Here we would definitely need more context.

[6]The parser used was, according to the information found, trained on the *Penn Treebank* for English which contains the Wall Street Journal corpus, The Brown Corpus (writing), The Switchboard Corpus and the ATIS Corpus (speech). By looking at the tagging results, it was evident that it was biased towards an analysis of nouns. This is why we decided to use the *TreeTagger* which gained much better results. Surprisingly, it was trained on the same corpus, although the tags differ. So far we haven't been able to solve this puzzle.

[7]The command to run the script is pc-compounder.sh -c.

below:

```
VBG     > She is
V       > He can
W       > I wanted to ask
UH      > He cried :
RP__TO  > He commanded :
IN      > He shouted :
.*      > I prefer the
```

If a tag string is matched, the corresponding context is inserted. The first rule inserts a context where the non-head's first element is a gerund as in *this "powdering my nose" act*:

```
VBG > She is
```

For a non-head beginning with a verb (that can be interpreted as imperative) as in our examples above (*"Say No To Strangers" campaign* or *"show the shirt" routine* the rule

```
V > He can
```

inserts a context which helps the parser to predict that this element really is a verb (and not a noun that follows a determiner).
For a non-head which is a (yes-no or constituent) question as in *"what's the point" vein* the respective context is inserted by the following rule:

```
W > I wanted to ask
```

For a non-head beginning with an interjection, the rule

```
UH > He cried:
```

inserts a context which makes it easier for the parser to predict that the non-head is an exclamation.
To correctly identify a non-head that begins with an adverb (or particle) as in *"down to earth" policy* the following rule inserts the respective context:

```
R[PB]__TO > He commanded:
```

To correctly identify the preposition as first element in our example *"up yours Vivien, fuck you Thatcher" explosion* the rule

```
IN > He shouted:
```

inserts the respective context.
If the contexts defined above are not matched the default is the following rule:

```
.* > I prefer the
```

This set of rules can be extended by adding them to the file pc-contexts.txt. The merit of this method is that non-heads of all types of PCs occurring in respective datasets can be correctly analysed, i.e. an accurracy of 100% can be reached.
Figure 7 is the flowchart for the method proposed here with the revisions made according to the results of the evaluation.

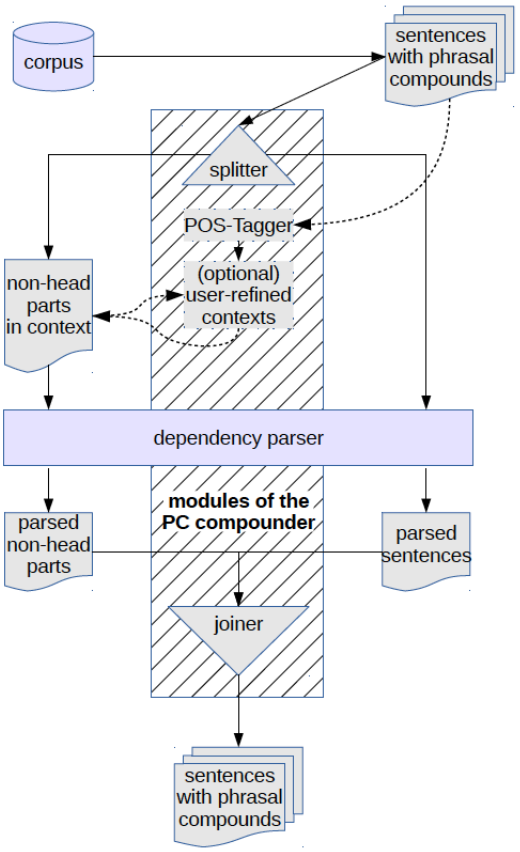

Figure 7: Flowchart for the proposed method

6. Conclusions

In this paper we have shown that a state-of-the-art dependency parser can be successfully used as a compounder to syntactically analyse phrasal compounds extracted from English corpora. The proposed method uses tools already available and therefore can be seen as an application of a "train once, use many" approach. The evaluation process revealed that despite the good accuracy achieved refinements were in order to improve the results. More precisely, in the module of the PC compounder we added a step which specified user-defined contexts which were sensitive to the part of speech of the non-head parts. To improve tagging results we added *TreeTagger*, another already available and published resource.
The method proposed can be extended to PCs where the elements of the non-head are marked with hyphens. In this case the step where two temporary CoNLL files are created can be reduced to one, since the addition of hyphens in tmp-pc-input.conll is not needed. In the case of PCs where the non-head is not marked at all, the method would still be successful because it is applied to a text file that contains the output of a search for PCs.
The more problematic point is to find such PCs in corpora in the first place[8]. One option would be to use the parsing

[8]Searching for PCs with quotations and hyphens we also al-

errrors identified in the evaluation process as detectors of non-marked PCs.

Instead of using an already available dependency parser, another possibility would be to improve the parser by retraining it on pc-output. Another option would be to use a specific dependency relation instead of PMOD. Since these issues were not the objective of the paper we leave it for further research.

7. Resources

The database of phrasal compounds as well as the Perl scripts used for the extraction/injection of embedded structures are available via the LRE Map.

References

Björkelund, A., Bohnet, B., Hafdell, L. and Nugues, P. 2010. "A high-performance syntactic and semantic dependency parser". In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, 33–36. Stroudsburg, PA, USA: Association for Computational Linguistics.

Bohnet, B. and Nivre, J. 2012. "A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing". In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1455–1465. Jeju Island, Korea: Association for Computational Linguistics.

Brown, R. 2002. "Corpus-driven splitting of compound words". In *Proceedings of the TMI-2002 conference, Keihanna, Japan, March 13-17*, 12-21.

Dađason, J. F. and Bjarnadóttir, K. 2014. "Utilizing constituent structure for compound analysis". In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 26.-31.5.2014*, Reykjavik, Iceland: European Language Resources Association (ELRA).

Haider, H. 2013. *Symmetry Breaking in Syntax*. Cambridge: Cambridge University Press.

McShane, M., Beale, S. and Babkin, P. 2014. "Nominal Compound Interpretation by Intelligent Agents". *LiLT* 10 (1).

Meibauer, J. 2003. "Phrasenkomposita zwischen Wortsyntax und Lexikon". *Zeitschrift für Sprachwissenschaft* 22: 153–188.

Nakov, P. 2013. "On the Interpretation of Noun Compounds: Syntax, Semantics, Entailment". *Natural Language Engineering* 1 (1): 1–40.

Schiller, A. 2005. "German Compound Analysis with wfsc". In *Proceedings of the Fifth InternationalWorkshop of Finite State Methods in Natural Language Processing (FSMNLP)*, 239–246. Helsinki, Finland.

Trips, C. 2012. "Empirical and theoretical aspects of phrasal compounds: against the "syntax explains it all" attitude". In *On-line Proceedings of the Mediterranean Morphology Meeting 8, Cagliari 2011, 'Morphology and the Architecture of Grammar'*, A. Ralli (ed), 322–46.

Trips, C. 2016. "The relevance of phrasal compounds for the architecture of grammar". In *The Semantics of Compounding*, P. ten Hacken (ed), 153–177. Oxford: Oxford University Press.

Trips, C. and Kornfilt, J., (eds). 2015. *Phrasal compounds from a typological and theoretical perspective*, volume 68 of *Special edition of STUF*. Berlin: De Gruyter.

ways checked whether we find the same non-heads without marking. The result was negative although, of course, we cannot totally negate their existence.