Akku•Bohr•Hammer vs. *Akku•Bohrhammer*: Experiments towards the Evaluation of Compound Splitting Tools for General Language and Specific Domains

Anna Hätty[1,3], **Ulrich Heid**[2], **Anna Moskvina**[2],
Julia Bettinger[1,3], **Michael Dorna**[1] and **Sabine Schulte im Walde**[3]
[1]Robert Bosch GmbH, Corporate Research
[2] Institute for Information Science and Natural Language Processing,
University of Hildesheim
[3]Institute for Natural Language Processing, University of Stuttgart
{anna.haetty,michael.dorna}@de.bosch.com,
{heidul,moskvina}@uni-hildesheim.de,
{julia.bettinger,schulte}@ims.uni-stuttgart.de

Abstract

We present a comparative evaluation study for splitting German compounds which belong to general language or to a specific domain. For the domain, we focus on DIY ("do-it-yourself"). The study consists of two parts: First, we evaluate three tools for compound splitting in German, one based on lexicons and corpus frequencies and two based on language-independent statistical processing. We introduce the tools, discuss the data and the construction of a gold standard, and show first results for binary and ternary noun compounds, as well as for the handling of non-splittable items. In a second experiment, we post-train one of the splitters with text data from the DIY-domain, and evaluate the splitting performance on domain-specific compounds.

1 Introduction

German is a highly compounding language, which means that several simple words like *Akku* "battery", *bohren* "to drill" and *Hammer* "hammer" are combined to form a complex word like *Akkubohrhammer* "cordless hammer drill". As a result, these complex compounds can be rather infrequent. In order to automatically process them, it is often useful to split them into their (usually more frequent) components, by using a compound splitter. However, compound splitting is a complex task, because there are often several splitting options possible. Splitting compounds which originate from specific domains further aggravates the problem: Both compounds and components might be even more infrequent, and a splitter might not have seen such data in the training stage, because it was trained on general language data.
For those reasons, we establish two evaluation settings to get a better insight into compound splitting for general language and for specific domains: (i) we compare several splitters with respect to their performance on both general language and domain-specific compounds and (ii) we post-train a splitter with domain data and evaluate the effect on domain-specific compounds.

In the first setting, we report on the comparative evaluation of three published tools. As a basis we use data from a specialized corpus, a general language corpus and the word formation literature. As the application domain is do-it-yourself instructions (DIY) from online forums, and we targeted the extraction and semi-automatic description of terminology candidates from the forum texts, compound splitting was mainly addressed with ontology building in mind; typically, heads of determinative noun compounds are hypernyms of such compounds. By splitting a noun like *Bandssäge* ("bandsaw") into *Band•säge*, the noun *Säge* can be identified as a hypernym of *Bandsäge*. Consequently, we only worked on noun compounds so far, even though adjective compounds would be equally interesting and even less covered by state of the art analyses of compound splitting. While split points are the main issue when it comes to the quality of the analysis of binary compounds, structure plays a major role for ternary compounds and items composed of more than three morphemes. Thus, for tri-morphemic compounds, we assessed both morpheme decomposition and structure assignment.

In the second setting, we post-train one of the compound splitters on a DIY text corpus. We then split all noun compounds in the corpus using the original and the modified splitter, and compare the results.

The paper is organized as following: In section 2 we will give an overview about the related work and in section 3, we introduce the three compound splitters. Section 4 describes the data that were used for the first experiment; additionally it gives details about how to create the compound gold standard, and how it can be used for evaluation. Section 5 describes the settings of the second experiment, how to post-train a splitter and which data were used. In section 6, we perform a detailed evaluation of the experiments. In section 7, we present and discuss aspects of the outcome of our evaluation, and in section 8, we conclude and point to needs with regard to future actions.

2 Related work

There exist a variety of compound splitters, which rely on different methodologies. There are linguistically motivated splitters, that rely on word frequencies (Koehn and Knight, 2003; Cap, 2014; Weller-Di Marco, 2017). CharSplit (Tuggener, 2016) however relies on a character-based method. A recent trend is to exploit distributional semantics to find the correct components (Ziering et al., 2016; Riedl and Biemann, 2016). Similarly, another splitter relies on semantic analogies (Daiber et al., 2015). Beside using different methodologies, the splitters return different splittings. For example, the Simple Compound Splitter by Weller-di Marco (2017) can return a binary or an n-ary split, lemmatize and POS-tag the components. CharSplit, however, does only a binary splitting. The output might depend on the application the splitter was designed for; for example, CharSplit was designed to find the compound heads in order to facilitate coreference resolution.

To our knowledge, no huge compound splitter comparison exists; Escartín (2014) conducts a small comparative study with two compound splitters. In addition, there is little work on domain adaptation of compound splitters. Macken and Tezcan (2018) perform Dutch compound splitting, and adapt the splitter to the automotive and the medical domain. They find that only using general language data performs better than only using domain-specific data, but a combination of both leads to the best results.

3 Tools for splitting German compounds and their evaluation

While a number of well-known and some upcoming tools for splitting German compounds exist, we are not aware of recent activities towards the comparative evaluation of the output quality of such tools. An older landmark for word formation evaluation of German as a whole is the Morpholympics contest, held in 1994 (Hauser, 1994). We briefly report about both, tools and evaluation.

3.1 Tools for compound splitting

In a general way, and especially with a view to the kind of evaluation we carried out, tools for compound analysis may be subclassified according to the kind of output they provide:

- tools only providing morpheme decomposition;
- tools providing morpheme decomposition and one or more structure proposals.

In addition, one may consider further types of tool output, e.g. category values of the morphemes identified. While this classification is based on the kinds of output produced by the tools, one may also distinguish symbolic vs. hybrid vs. purely statistical, machine learning based tools, according to the approach. In the following, we briefly describe the tools we analyzed, and we mention a few more that may be used in a second round of the evaluation.

3.2 SECOS: Unsupervised Compound Splitting With Distributional Semantics

Unlike most systems that rely on dictionaries or are trained in a supervised fashion, SECOS (Riedl and Biemann, 2016) relies entirely on distributional semantics. The hypothesis investigated by the researchers postulates that compounds are similar to their constituting word units. Their method is based on a distributional thesaurus that is computed using a tokenized monolingual background corpus without any additional linguistic processing. The first step is the extraction of a candidate word list that defines the possible word units of compounds. The second step is splitting the compounds. The last step is a ranking of the splits and returning the top-ranked ones. The method is proven to be language independent: several experiments were conducted on German and on Dutch, they produced equally good results. The tool is freely available.[1]

[1] https://github.com/riedlma/SECOS.

3.3 Compound splitting tool from Tübingen University

The authors (Ma et al., 2016) introduced a letter sequence labelling approach, which can utilize rich word form features to build discriminative learning models that are optimized for splitting. The prediction of labels is achieved by training conditional random fields. The method is language-independent and does not require any linguistical preprocessing. Splitting is conducted at the surface form level. The current system, available for testing, is trained to split multi-constituent compounds at the boundaries of all the constituent words, instead of only splitting at the top level (complete morpheme decomposition).

3.4 CompoST: Compound Splitting Tool

The tool splits compounds into their morphemes using morphological rules and corpus frequencies. The underlying method (Cap, 2014) involves using the geometric mean of subword frequencies to disambiguate possible splits. CompoST was developed for compound processing in statistical machine translation, but it can equally be used as an independent module for morphological analysis. It requires frequency counts derived from a corpus; candidate items are analysed by SMOR (a rule based morphological analyser for German) (Schmid et al., 2004). CompoST allows to set different parameters and therefore to gain different versions of output. For instance, it can split a word even when frequency scores suggest that the word can not or should not be split (forced splitting), or it can split only nouns. One of the drawbacks of the tool is that words unknown to SMOR cannot be split, as well as disambiguation of possible splits is entirely based on frequency, and this might lead to inconsistencies on a non-lemmatized word list.

4 Gold standard for compound splitting

A gold standard evaluation was carried out, in the framework of our project on term candidate extraction from do-it-yourself instructions (DIY). While the focus of the evaluation was on the coverage of the data from the DIY-corpus, and on the quality of the respective analyses, we also wanted to explore the performance of the tools on general language data. We created a database that contains the gold standard, as well as the output of individual tools. In this way, all elements of the evaluation can later be enhanced: more gold data can be added, and the results of further tools can be compared.

4.1 Sources and selection criteria

For both, specialized and general language, corpus data were used, but with different objectives. For specialized language, we used a corpus of 11 million running words, composed of expert texts and user generated content (=UGC) from the domain of DIY instructions. The relationship between expert and UGC texts was roughly 1:5. For the gold standard, we extracted noun compounds (by means of TreeTagger-assigned pos="NN" annotations) from three frequency bands: top, medium and low frequency items. Given the overall frequency distribution of nouns, the distribution of candidate items shown in Table 1 was achieved.

We are aware that the "medium" frequency band is as yet underpopulated. Additional sampling may be needed to provide roughly the same quantities of data as for the two other frequency bands. However, this would not even out the relationship between binary and trimorphemic candidates, which is uneven as well but likely relatively close to the distribution to be expected in the texts under analysis. To counterbalance the almost proportional sampling from the specialized corpus, we added data from general language materials. In this part of the gold standard, we did not aim at replicating frequency distributions from a given corpus, but we rather targeted a collection of all cases that are discussed as relevant in the literature on German compounding. This approach is similar to part of Hauser's (1994) sampling method. Thus ca. 200 items were taken from the standard handbook on German morphology by Fleischer and Barz (1995). We cross-checked however the chosen items against 200 M words of news texts and against the SdeWaC corpus (Faaß and Eckart, 2013), and only used items present in at least one of them. These items provide a wide range of possible issues for compound splitting, e.g. adjectival non-heads that are not in the positive form (*Mehrarbeit* "additional work"; *Reinststoff* "ultrapure substances", lit.: "ultrapurest substances") or compounds with phrasal non-heads (*Heißwasserspeicher* "boiler", lt.: "hotwater storage").

4.2 Annotation of the gold standard

The annotation was carried out manually, by one linguist. The reason why we consider this sufficient is that the underlying guidelines are based on standard analyses from morphological theory

frequency range	**frequency**	**non-split**	**binary**	**trimorph.**	**total**
top	f > 100	44	329	67	440
medium	41 > f > 37	6	113	29	148
low	f=12	21	312	100	433
total		71	754	196	1,021

Table 1: Frequency-based sampling of noun compounds from an 11 M word corpus of DIY forum texts.

(Ortner et al., 1991; Pümpel-Mader et al., 1992; Fleischer and Barz, 1995; Donalies, 2011; Donalies, 2014); for items which, according to these sources, can receive more than one analysis, all valid analyses were included in the gold standard, such that tools providing one of them were not punished. The annotated data were stored in a database. The following features were annotated:

- split points on the form level - in the sense of Koehn and Knight (2003) - and lemma forms of the morphemes;
- pos categories of the non-head morphemes;
- structure of tri-morphemic compounds (left vs. right branching).

In addition, the following documentary data were annotated by automatic means:

- number of split points (for easy counting of over- and undersplitting cases);
- lemma frequency of the item tested, as well as of its components in 200 M words of news text and in SdeWaC.

The following is a simplified example of the linguistic representation of the items in the gold standard database; the first feature is the POS combination of the non-head morphemes; it is followed by the lemma from the corpus, its decomposition into morphemes at the level of surface forms, its topmost split at the level of surface forms, as well as the morpheme decomposition and the structure proposal (=topmost split) on the level of lemmas.

```
adj-v Kleinstlebewesen
– kleinst lebe wesen + kleinst
  Lebewesen
– klein leben Wesen + klein
  Lebewesen
```

The double annotation, at both lemma and surface level, ensures compatibility with most types of tool outputs and thus eases the comparison.

4.3 Data annotated

As mentioned above, we included noun compounds of three kinds in the database: binary and tri-morphemic compounds, but also items that cannot be split, e.g. because they are derivation products. We also included ca. 30 items which allow for two structural analyses, e.g. *Meerwasserentsalzungs•Anlage* vs. *Meerwasser•Entsalzungsanlage* ("desalination plant", lit.: "sea water desalination plant"). The distribution over the full data set is given in table 2.

frequency range	**#**
non-splittable	86
binary	
- N+N, Adj+N	715
- V+N	118
tri-morphemic	294
total	1,239

Table 2: Distribution of compounds over the full data set.

5 Post-training with domain-specific text data

Adapting a compound splitter to a certain domain of interest, as DIY in our case, might improve the compound splitting for two reasons: First, the domain-specific components of a compound might be infrequent in general language, and that is why the correct split or base form of the component cannot be found. For example, the compound *Eloxierverfahren* ("anodizing procedure") should be splitted and lemmatized to *eloxieren•Verfahren* ("to anodize•procedure"). Secondly, splitting probabilities might be skewed because a certain split is more likely in general language, while another one is more likely within the domain. For example, the compound *Rohrverbinder* ("pipe connector") is likely to be split as *Rohr•Verb•Inder* ("pipe•verb•Indian") in general language, because the three components do occur more often in general language than the correct components *Rohr* ("pipe") and *Verbinder* ("connector").

However, post-training of a compound splitter on a domain-specific corpus is not always possible. It depends on the design of the tool and if the original training data are available for updating.
We adapt the splitter CompoST. CompoST relies on frequency counts derived from a corpus, in the default case a general-language corpus. To adapt the splitter to the DIY domain, we compute all the frequency counts for a DIY text corpus. Then we either add the frequencies to existing token entries, or create new ones. We use a domain-specific DIY corpus with 5.6 million words. The texts were collected from different sources, but all of them are DIY-related. There are texts produced by domain experts as well as by interested lay users, such as encyclopedia texts, DIY-instructions and manuals. Preprocessing has been done with SpaCy[2] (Honnibal and Johnson, 2015). Working with the German language model of SpaCy, we make use of the tokenizer, the POS-tagger and the lemmatizer. While the tagging itself is based on a convolutional neural network, the lemmatizer still works with a conservative look-up table. We use the POS-tags to select noun compounds as candidates for compound splitting.

6 Evaluation

6.1 Comparison of compound splitters

6.1.1 Evaluation methods

We mainly follow Koehn and Knight's (2003) procedures for the comparison of our gold standard splits with the output produced by the tools. To ease the quantitative assessment of over- and undersplitting, we count the number of split points in each gold standard item and in each tool output for the respective item and annotate this number back into the database. As we offer the gold analyses both on word forms and on lemmata, we use both versions as alternatives to match the tool output against: the results of each tool (or of each version of tool output) are inserted, for each gold item, into the respective row of the database table; for each tool output, the table is thus enlarged by one or several complete column(s). Not all tools provide just the split points; some provide in addition pos-features or other descriptive output. When preprocessing the tool output we keep track of such specificities. We evaluated the analyses provided by the tools in terms of correct vs. incorrect split points, over- and undersplitting. Later, we will include an evaluation with regard to POS categories of the components wherever possible.

6.1.2 Results

According to the proposed methodology the first assessment of tool quality is achieved by a simple comparison of the output in the terms of:

- correct splits (when the splits provided by the tool either correspond to the morphological or structural gold splits, for example: *Bienenwachslasur* will result in the following gold splits: *Bienen•wachs•lasur* and *Bienenwachs•Lasur*);
- incorrect non-splits (when the tool perceives a word as a non-compound, a special form of undersplitting);
- wrong split points.

In this paper we present the result of such an analysis only for N+N type compounds (*Gerölllawine, Bombengeschäft, Tagblatt*), as well as for V+N type compounds (*Isolierschlauch, Meldeeinheit, Schleifgerät*), and also for certain types of tri-morphemic compounds (*Sperrholzrest, Heizkörpernische, Heißklebepistole*). The results obtained for binary compounds, N+N type (N = 626), are listed in Table 3.

Though CompoST clearly outperforms the other tools, some nouns still remain unsplit. Nevertheless it also made fewer wrong splits than SECOS or the TU-tool. The latter is almost as good as CompoST in terms of undersplitting, though it produced almost twice as many wrong splits. While SECOS made less mistakes with split points than the TU-tool, it was not as good as in distinguishing compounds from non-splittable items. One of the reasons for this performance might be the specialised nature of the data, as most of the N+N type compounds came from the domain of DIY instructions, such as: *Steinbearbeitung, Bohrmaschine, Drehzahl*. The results obtained for binary compounds of the V+N type (N = 118) are presented in Table 4.

In this case CompoST produced more nonsplits than the other tools, though its general performance is still higher than 65%, and only one compound was wrongly split (*Wegwerfgesellschaft*:

[2] https://spacy.io/

Tool	correct	non-split	wrong split
CompoST	582 (93%)	9 (1,4%)	35 (5,6%)
TU-tool	500 (80%)	15 (2,3%)	111 (17,7%)
Secos	496 (79%)	50 (7,8%)	79 (13,2%)

Table 3: Quantitative results on N+N compounds.

Tool	correct	non-split	wrong split
CompoST	78 (66%)	39 (33%)	1 (1%)
TU-tool	92 (78%)	2 (1,7%)	24 (20,3%)
Secos	75 (63,7%)	19 (16%)	24 (20,3%)

Table 4: Quantitative results on V+N compounds.

wrongly split as *??Weg•Werf•Gesellschaft* instead of *Wegwerf•Gesellschaft*). The undersplitting tendency observed in N+N type compounds can be detected here as well. However the TU-tool outperforms the others with almost 78% of correct splits. The TU-tool and SECOS share the ca. 20% of wrong splits (*??Ein•Lege•Bretter* (TU-tool) and *??Einlegebre•Tter* (SECOS) instead of *Einlege•Bretter, ??Unter•Legscheibe* (TU-tool) and *??Unter•legscheibe* (SECOS) instead of *Unterleg•Scheibe, ??Ans•Aug•Leistung* (TU-tool) and *??Ansau•Gleis•Tung* (SECOS) instead of *Ansaug•Leistung*). Examples of selected ternary compounds of different types (N = 173) are given in the table 5.

There may not be enough candidate data to assess all patterns, as A+N+N and V+N+N are rather rare in our texts; more data may be needed in the future to allow us to come up with a more meaningful evaluation. Nevertheless, both the TU-tool and SECOS provided consistently good results, with low percentages of wrong splits and almost no undersplitting. CompoST on the other hand exhibits a considerable amount of undersplitting, but produces only very few wrong splits. It remains unclear why A+N+N compounds lead to problems with CompoST. Our test set contained also non-compounds (N = 86), so that we could investigate oversplitting and the ability to distinguish compounds from other word formation products. The non-splittable candidates are mostly derivatives, some of which are phrasal derivatives:

- Derivation products: *Möglichkeit, Verschraubung*;
- Phrasal derivatives: *Rechtwinkligkeit*

The results are presented in Table 6.

Again CompoST clearly outperforms other tools in this task. It provides many good solutions and only a small amount of errors. Both the TU-tool and SECOS tend to produce erroneous splits in almost two thirds of the cases; their recognition capacity of non-splittable terms is thus not particularly good yet. All the three systems presented above were tested and their output was analyzed. Due to the underlying processing method the TU-tool and SECOS more often produce oversplitting of compounds (SECOS: *??W•Ärmer•Ückgew•Innungs•Anlage* instead of *Wärme•Rückgewinnungs•Anlage, ??Wasser•Rückgew•Innungs•Anlage* instead of *Wasser•Ruckgewinnungs•Anlage*, and *??Un•Kennt•Lich•Machung* instead of *Unkenntlichmachung*; TU-tool: *??Ver•Blend•Mauer•Werk* instead of *Verblend•mauerwerk*, and *??Sch•Werst•Behinderten•Betreuung* instead of *Schwerst•Behinderten•Betreuung*), while CompoST undersplits compounds from the general language even when the parameters are set to enforce splitting.

6.2 Post-training on domain-specific text data

For the evaluation of post-training CompoST, we take all word types from the DIY corpus as candidates for compound splitting, which are tagged as nouns. We both run the original CompoST (ORIG) and the version of CompoST adapted to the DIY domain (MOD). The results are shown in table 7. Overall, the modified version of CompoST finds more compounds than original CompoST does (first two rows of table). However, the difference is not big (259 compounds). Furthermore, for the majority of the cases, both splitter

Type	Tool	correct	non-split	wrong split
N + N + N	CompoST	97 (85%)	14 (12,3%)	3 (2,7%)
(114)	TU-tool	105 (92%)	0 (0%)	9 (8%)
Span●Holz●Platte	Secos	92 (81%)	3 (2,7%)	19 (16,3%)
A + N + N	CompoST	11 (31,4%)	21 (60%)	3 (8,4%)
(35)	TU-tool	31 (89%)	0 (0%)	4 (11%)
Rund●holz●stab	Secos	30 (86%)	0 (0%)	5 (14%)
V + N + N	CompoST	22 (88%)	3 (12%)	0 (0%)
(25)	TU-tool	22 (88%)	0 (0%)	3 (12%)
Senk●kopf●schraube	Secos	21 (84%)	0 (0%)	4 (16%)
All types	CompoST	195 (66%)	91 (31%)	8 (3%)
(294)	TU-tool	261 (89%)	3 (1%)	30 (10%)
	Secos	234 (80%)	8 (3%)	52 (17%)

Table 5: Quantitative results for selected ternary candidates.

Tool	correct	wrong split
CompoST	82 (95%)	4 (5%)
TU-tool	33 (38%)	53 (62%)
Secos	43 (50%)	43 (50%)

Table 6: Quantitative results on non-splittable items.

versions split identically (row 3), i.e. roughly 95% of the compounds split by MOD are split in the same way by ORIG. Rows 4 to 9 show the cases where the splitters do not agree, which is further analyzed below.

feature	#
all ORIG splits	59,936
all MOD splits	60,195
same split	57,145
only MOD splits	640
only ORIG splits	411
MOD more splits	232
ORIG more splits	227
different split points	127
lower/upper difference	1,793

Table 7: Comparison of the splitting results for the original CompoST (ORIG) and CompoST post-trained on a DIY corpus (MOD).

Only MOD splits vs. only ORIG splits. MOD splits more compounds than ORIG. In return, it misses compounds which were originally split (“only ORIG splits”). This makes up roughly 2/3 of the size of the compounds only split by MOD. It seems likely that the missed compounds originate from general language, and the newly split ones are domain-specific. However, when analyzing the compounds, this is not the case; clear DIY-compounds like *Akkuschrauber* (“screwdriver’)’, *Stichsäge* (“padsaw”) or *Heimwerker* (“DIYer”) are not split by MOD.

Secondly, we want to analyze the impact of hyphenated compound candidates. An example would be *Douglasien-Bodendielen* (“douglas fir-floor boards”), where the split point is obvious because of the hyphen. There are rare cases where such a split would be wrong, e.g. *3-in-1* or *200-er*. We throw out all compounds where the split point is set at the hyphen and show the result in table 8 (columns “only X splits”). Obviously, most compounds that MOD missed were hyphenated compounds; for closed compounds, MOD shows a superior performance for both binary and ternary compounds.

	only X splits		X more splits	
	ORIG	MOD	ORIG	MOD
binary	43	600	-	-
ternary	0	50	137	22
nary	-	-	9	0

Table 8: Difference of splitting results for the original CompoST (ORIG) and post-trained CompoST (MOD) with disregarding all compounds with splits at hyphens.

MOD more splits vs. ORIG more splits. In these cases, both splitters split the same compound but the number of splits is different. While for the overall results (table 7) this part seems to be rather equally sized for the splitters, focusing on the closed, not hyphenated compounds again (table 8, columns "X more splits") the picture is quite different. MOD produces fewer splits, i.e. contracts components within a compound. For example, ORIG splits *Schraubendreherklingen* ("screwdriver blades") as *Schraube•Dreher•Klingen* ("screw•driver•blades"), while MOD splits *Schraubendreher•Klingen* ("screwdriver•blades"). We conclude that MOD finds some compounds to occur frequently and thus does not split them anymore. This intuition also coincides with the results from the previous paragraph, that DIY compounds like *Akkuschrauber* ("screwdriver") are not split anymore by MOD.

Different split points. In these cases, both splitters split the same compound and return the same number of splits, but the split points are differently set. When analyzing the compounds, we find that in most cases the results are different because the modifier is either lemmatized as noun or verb, e.g. *Putz/putzen* ("plastering/to clean"), or the lemma is different: *Dosen* → *Dose/Dosis*. Some errors result from the Fugen-s (*Prozessor•Steuerung* "processor controlling" vs. ??*Prozessor•Teuerung*, lit.: "processor increase in prices"), or a completely wrong split. MOD performs superiorly to ORIG because it always selects the more likely lemma in the domain (e.g. *Putz* instead of *putzen*). We randomly select 30 compounds of this category and compare the splitting results; MOD splits 18 times correctly, ORIG only 8 times (in the other cases, both splits were incorrect).

Lower/upper difference. In these cases, both splitters split the same compound, return the same number of splits and find the same split points. Only upper- and the lowercasing is different. When analyzing the respective compound splits, one can see that it is mostly again the modifier which is different. Sometimes this is a discrepancy between verb and nominalized verb (e.g. *Sägetisch* "sawing table" is either split as *sägen•Tisch* "to saw•table" or *Sägen•Tisch* "sawing•table"), or upper- or lowercasing is just wrong (e.g. *Nahtkontrolle* is split as *naht•Kontrolle* "joint examination"). It is unclear where this effect comes from. When again extracting 30 compounds randomly, MOD lemmatizes 15 times correctly, and ORIG lemmatizes 14 times correctly. To conclude, no splitter shows superior performance here.

7 Discussion

In general, it is rather difficult to compare and evaluate the performance of different compound splitters. They return diverse splittings, e.g. they either return binary or n-ary splits, lemmatize the results or additionally POS-tag them. For some splitters, there even are several settings available (as for example, restricting either to a binary split or allowing an n-ary split). Thus, sometimes a comparison can be hard. For example, do we prefer a splitter that does not lemmatize against a splitter that lemmatizes, but sometimes returns wrong lemmas? Finally, the follow-up task for the compound splitting might decide which splitter we will use.

8 Conclusion and outlook

We presented a two-part study to evaluate the performance of German compound splitters on noun compounds, for general language and for specific domains. In a first experiment, we conducted a gold-standard-based evaluation of three compound splitters on general-language and domain-specific compounds. The splitters are CompoST, SECOS and a CRF-based tool from University of Tübingen. We explained data sampling from specialized corpora and from an inventory of general language phenomena in compounding. We noted that CompoST tends to undersplit compounds (likely due to a lack of lexical knowledge in SMOR), while the other two tools tend to oversplit. Consequently, CompoST also performs best on non-splittable items (95% correct vs. 50% for the second best tool). Its precision is highest for N+N compounds. TU-Tool produces more correct splits on V+N compounds, but also produces more incorrect splits. It is the best-performing tool on tri-morphemic noun compounds, with SECOS being second and CompoST last (only 66% correct vs. 89% with TU-Tool). TU-Tool produces a slightly higher amount of wrong splits than CompoST for tri-morphemic compounds, but therefore CompoST does not split nearly one third of the compounds. In general, CompoST rarely produces splits the result of which are non-morphemic letter sequences (in contrast to *Einlegebre·Tter* discussed in section 6.1.2).

In a second experiment, we post-trained CompoST on domain-specific DIY data, and compared the results for splitting domain-specific compounds. We found that for roughly 95% of the compound candidates, the original and the modified splitter return identical splits. For the rest of the compounds, we performed a detailed evaluation with respect to several features, like the number of splits or a difference of the exact split points. We find that in these cases the adapted CompoST mostly outperforms the original one, especially for binary and ternary closed compounds. This qualitative improvement is quantitatively watered down by the fact that the original CompoST more often splits hyphenated compound candidates than the post-trained version. The modified version more often contracts components within an n-ary compound, presumably due to the increased number of occurrences of a complex component (e.g. *Heimwerker*) in the data used for post-training.

Overall, the comparison of compound splitters proved to be more difficult than one would expect, as the tools come with widely diverging features: some tools only provide one split-point, others do not come with training data, yet others include lemmatization of the output, which in some cases can be a source of further errors. Against this background, we see a need for further detailed methodological work on the topic.

References

Fabienne Cap. 2014. *Morphological processing of compounds for statistical machine translation.* Ph.D. thesis.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. 2015. Splitting compounds by semantic analogy. In *Proceedings of the 1st Deep Machine Translation Workshop*, pages 20–28, Praha, Czechia. ÚFAL MFF UK.

Elke Donalies. 2011. Basiswissen deutsche wortbildung. 2., überarbeitete auflage. *Tübingen/Basel: Francke.*

Elke Donalies. 2014. Morphologie: Morpheme, wörter, wortbildungen. *Ossner, Jakob/Zinsmeister, Heike (Hrsg.): Sprachwissenschaft fr das Lehramt*, pages 157–180.

Carla Parra Escartín. 2014. Chasing the perfect splitter: A comparison of different compound splitting tools. In *LREC*, pages 3340–3347.

Gertrud Faaß and Kerstin Eckart. 2013. Sdewac–a corpus of parsable sentences from the web. In *Language processing and knowledge in the Web*, pages 61–68. Springer.

Wolfgang Fleischer and Irmhild Barz. 1995. *Wortbildung der deutschen Gegenwartssprache.*

Roland Hauser. 1994. Results of the 1. morpholympics. LDV-FORUM.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *10th Conference of the European Chapter of the Association for Computational Linguistics.*

Jianqiang Ma, Verena Henrich, and Erhard Hinrichs. 2016. Letter sequence labeling for compound splitting. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 76–81.

Lieve Macken and Arda Tezcan. 2018. Dutch compound splitting for bilingual terminology extraction. *Multiword Units in Machine Translation and Translation Technology*, 341.

Lorelies Ortner, Elgin Müller-Bollhagen, Hanspeter Ortner, Hans Wellmann, Maria Pümpel-Mader, and Hildegard Gärtner. 1991. *Deutsche Wortbildung. Typen und Tendenzen in der Gegenwartssprache. Vierter Hauptteil: Substantivkomposita.* Berlin, New York: De Gruyter.

Maria Pümpel-Mader, Elsbeth Gassner-Koch, Hans Wellmann, and Lorelies Ortner. 1992. *Deutsche Wortbildung: Typen und Tendenzen in der Gegenwartssprache; eine Bestandsaufnahme des Instituts für Deutsche Sprache, Forschungsstelle Innsbruck. Hauptteil 5. Adjektivkomposita und Partizipialbildungen.* Berlin, New York: de Gruyter.

Martin Riedl and Chris Biemann. 2016. Unsupervised compound splitting with distributional semantics rivals supervised methods. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 617–622.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. Smor: A German computational morphology covering derivation, composition and inflection. In *LREC*, pages 1–263. Lisbon.

Don Tuggener. 2016. *Incremental coreference resolution for German.* Ph.D. thesis, Universität Zürich.

Marion Weller-Di Marco. 2017. Simple compound splitting for German. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 161–166.

Patrick Ziering, Stefan Müller, and Lonneke van der Plas. 2016. Top a splitter: Using distributional semantics for improving compound splitting. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 50–55.