Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4359–4367
Marseille, 11–16 May 2020

A Domain-Specific Dataset of Difficulty Ratings for German Noun Compounds in the Domains DIY, Cooking and Automotive

Julia Bettinger[1,2], Anna Hätty[1,2], Michael Dorna[1], Sabine Schulte im Walde[2]
[1]Robert Bosch GmbH, Corporate Research, Renningen, Germany
[2]Institute for Natural Language Processing, University of Stuttgart, Germany
{julia.bettinger, schulte}@ims.uni-stuttgart.de, {anna.haetty, michael.dorna}@de.bosch.com

Abstract

We present a dataset with difficulty ratings for 1,030 German closed noun compounds extracted from domain-specific texts for do-it-ourself (DIY), cooking and automotive. The dataset includes two-part compounds for cooking and DIY, and two- to four-part compounds for automotive. The compounds were identified in text using the *Simple Compound Splitter* (Weller-Di Marco, 2017); a subset was filtered and balanced for frequency and productivity criteria as basis for manual annotation and fine-grained interpretation. This study presents the creation, the final dataset with ratings from 20 annotators and statistics over the dataset, to provide insight into the perception of domain-specific term difficulty. It is particularly striking that annotators agree on a coarse, binary distinction between easy vs. difficult domain-specific compounds but that a more fine-grained distinction of difficulty is not meaningful. We finally discuss the challenges of an annotation for difficulty, which includes both the task description as well as the selection of the data basis.

Keywords: Domain-Specific Terminology, Term Difficulty, Closed Compounds

1. Introduction

Domain-specific texts contain domain-relevant vocabulary, which is potentially difficult to understand for people without specialized knowledge about the domain. The addressed audiences of domain-specific texts might also vary. Some texts are clearly written for experts while others are written especially for laypersons. It is thus important to identify the difficulty of words contained in domain-specific texts, as one of the building blocks to draw conclusions about domain-specific text difficulty.

For these reasons, we create a German closed noun compound dataset that was extracted from three domains: do-it-yourself (DIY), cooking and automotive. We chose closed compounds because they represent a kind of multiword expression that is easy to identify in text, and we therefore do not need elaborate preprocessing for finding valid phrases. At the same time, it is highly likely for multiword expressions in domain-specific texts that they carry domain-relevant meanings. For example, Justeson and Katz (1995) find that the most common form of a domain-specific term for English is a two-word noun compound.

In the following, we describe the development of the domain-specific compound dataset and its annotation for difficulty. This includes the construction of domain-specific corpora to extract the compounds (section 4.), and the actual creation of the compound dataset (section 5.). After evaluating the interannotator agreement (section 6.), the final gold standard is described (section 7.). We present statistics and insights about the dataset (section 8.) and discuss the challenges of annotating difficulty (section 9.).

2. Related Work

Detecting a lay reader's familiarity or difficulty with domain-specific expressions is a niche research area, and a subtask of the more general areas of complex word identification and domain-specific text readability assessment. It often involves subsequent steps of term substitution through simpler synonyms and providing an explanation (Elhadad, 2006; Kandula et al., 2010). Most studies focus on biomedical or medicals areas and the assessment of difficulty of domain-specific terminology. Approaches to evaluate familiarity prediction systems are diverse. Bouamor et al. (2016) rely on English Consumer Health Vocabulary that is included in the UMLS Metathesaurus (Zeng et al., 2007), whose vocabulary distinguishes between lay and specialized terms. Grabar et al. (2014) create a gold standard with manual annotations on a three-position scale: understand – partly understand – don't understand. Vydiswaran et al. (2014) perform a post-hoc evaluation of their presented models, letting a medical expert review a sample of 100 pairs, which were previously extracted as 'consumer' and 'professional' terms. Zeng-Treitler et al. (2008) measure a lay person's familiarity with a term based on the percentage of annotators who identify the term correctly.

3. German Closed Noun Compounds

Closed compounds consist of at least of two words, contracted together to form a compound without delimiting space, sometimes linked with a hyphen. Closed compounds are highly common in German. Two-part compounds consist of a *modifier*, the first constituent in German, and the morphological *head*, the second constituent in German. For noun compounds, the head and also the whole compound are nouns, for example:

Kartoffelsalat$_{Noun}$ ("potato salad")
modifier: Kartoffel$_{Noun}$ + **head**: Salat$_{Noun}$

Kochtopf$_{Noun}$ ("cooking pot")
modifier: kochen$_{Verb}$ + **head**: Topf$_{Noun}$

Weißbrot$_{Noun}$ ("white bread")
modifier: weiß$_{Adj}$ + **head**: Brot$_{Noun}$

4. Creation of Domain-Specific Background Corpora

Three domains are selected to collect corpus data for compound extraction: DIY, cooking and automotive. We select the cooking domain because a large amount of text data are available: recipes, ingredient and technique descriptions, and more, crawled from `kochwiki.org`, `wikihow.de`, `wikibooks.de` and related Wikipedia articles. For DIY we had a corpus already available, containing online texts mostly crawled from the BOSCH empowered homepages `bosch-do-it.de` and `1-2-do.com`. The corpus consists of user-generated content as well as expert texts (e.g. tool manuals, books on handicraft), and we further add material from `wikihow.de`. Finally, we choose the automotive domain because it contains particularly many technical terms. Texts are again crawled from Wikipedia and `wikihow.de`, and further we take the contents of an automotive handbook. For all domains, Wikipedia is crawled recursively by categories. The Wikipedia categories are manually filtered for categories which are contentwise too far away, as a further data cleaning step to maintain the topical focus of the corpora. Finally, all corpora are reduced to the size of the smallest corpus, which results in equally-sized corpora of 5.6 million tokens. The texts are tokenized, lemmatized and tagged with spaCy[1]; we applied lemma correction.

5. Creation of Compound Dataset

5.1. Extracting Compounds from Domain Corpora

All compounds in the texts that were POS-tagged as nouns are identified and extracted by the *Simple Compound Splitter* (SCS, Weller-Di Marco (2017)). We chose the SCS over other compound splitters because of its capabilities that were especially suited for our task: All components in the compounds get lemmatized and POS-tagged, and the splitter is capable of doing both binary and multiple splits. The SCS splitter was directly trained on the domain-specific corpora. The number of extracted compounds per domain is given in table 1. We mainly focus on two-part compounds, but due to the high number of longer compounds in the automotive domain (and expecting these to be highly technical), we also extract three-part and four-part compounds for that domain. However, in later processing steps, we will treat them as two-part compounds and only split them at the main split point.

domain	components	frequency
cooking	2	42,484
DIY	2	45,724
automotive	2	81,323
	3	73,675
	4	5,681

Table 1: Compounds extracted by the SCS splitter.

Table 1 shows that more two-part compounds are extracted for the automotive domain than for DIY and cooking. This is in line with our observation that automotive is the most technical domain, and with Clouet and Daille (2014), that "*[compounding] is particularly productive in specialized domains because of the necessity to denote the domain concepts in a very concise and precise way*" (p. 11).

5.2. Balancing and Filtering

Since the set of retrieved compounds is too large to be annotated completely, we select a balanced subset. We consider the following compound characteristics as relevant for our task:

- **frequency** of compound and components: How often do they occur in the respective domain-specific corpus as an independent unit (i.e. the components are not embedded within other words)?
- **productivity** of the modifier and head: In how many compound types does a certain modifier/head occur as a modifier/head?

Concretely, we then choose the following four criteria for balancing:

- frequency of the compound
- productivity of the head
- productivity of the modifier
- frequency of the head

Before balancing, we exclude all terms with a frequency smaller than three, because the annotators would be given three sentences for each term. This results in a pool of 12,400 cooking compounds, 16,935 DIY compounds and 20,468 automotive compounds. The set is balanced by dividing it into tertiles, i.e. dividing the set into groups of *low*, *mid* and *high* frequency and productivity, resulting in a total of $3^4 = 81$ classes. Then compounds are randomly selected from each class, and two annotators checked if the compounds are valid and split correctly. We further randomly inject a small amount of compounds which we find difficult, to counteract against the presumed imbalancedness of the dataset in favour of easy compounds. The final numbers of selected compounds for the gold standard are given in table 2.

domain	components	frequency
cooking	2	243
DIY	2	243
automotive	2	243
	3	162
	4	139
total		1,030

Table 2: Final gold standard set of compounds.

[1] `https://spacy.io/`

5.3. Annotation

The final dataset is rated by 26 annotators in total. The annotators are shown the highlighted compound accompanied by three domain-specific sentences. Based on the example sentences, they are asked to rate the compound type on the following Likert-like scale (Likert, 1932)[2]:

1: The term does not require any specialized knowledge
in order to be understood.

2: The term requires little specialized knowledge
in order to be understood.

3: The term requires specialized knowledge.
Parts of its meaning can be inferred from context.

4: The term requires specialized knowledge.
Its meaning cannot be inferred from its context.

6. Evaluation

6.1. Interannotator Agreement Measures

To evaluate our annotation we calculate three agreement measures: Fleiss' κ (Fleiss, 1971), the Jaccard index (Jaccard, 1902) and Spearman's ρ (Siegel and Castellan, 1988). Fleiss' κ is an extension of Cohen's κ (Cohen, 1960) for more than two annotators. The Jaccard index is calculated pairwise for all combinations of annotators. However, both these measures do not take the ranks into account. This means that they do not differentiate between a disagreement of 1 vs. 2 and 1 vs. 4. However, for our purpose this should be considered which is why we also calculate Spearman's ρ pairwise for all combinations of annotators. All measures are calculated over all domains, and in addition for each domain individually.

6.2. Selecting n-best Annotations

We decided to exclude one annotator in advance, because the annotator misunderstood the task and assigned a value to each sentence rather than to each compound type. The pairwise Spearman's ρ correlations for the remaining 25 annotators over all three domains are visualized in Figure 1.

We carry out the scaling down of annotators as follows:

1. Calculate the average over the pairwise ρ scores of each annotator with all other annotators.
   - Over all domains.
   - For each domain individually.

   This gives us four values per annotator.

2. Exclude annotators which have the least average agreement in the majority of the four cases.
   - If two annotators have lowest agreement in exactly half of four cases, we exclude both of them.
   - If for each of the four cases a different annotator has the lowest average agreement, we exclude the annotator with the largest difference to the second least agreeing annotator for any of the cases.

With this procedure the overall Spearman's ρ correlations develop as shown in table 3.

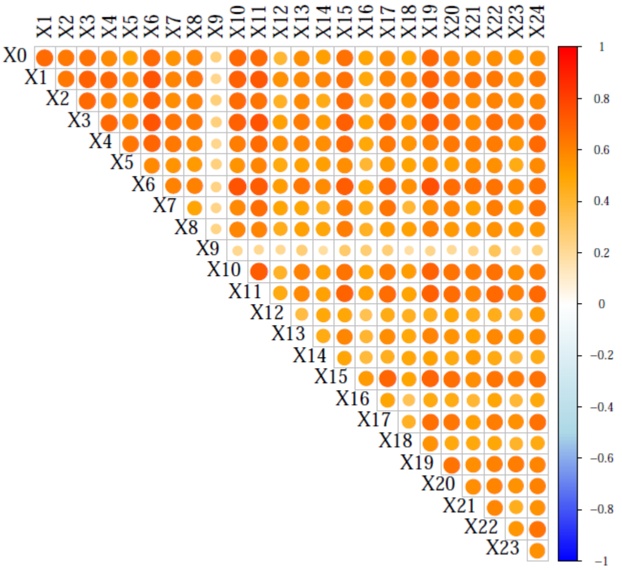

Figure 1: Pairwise Spearman's ρ correlations for 25 annotators over all domains.

#Anno.	Cooking	DIY	Auto.	All
25	0.5035	0.5093	0.5716	0.5449
24	0.5336	0.5463	0.5878	0.5714
23	0.5508	0.5569	0.5945	0.5824
21	0.5723	0.5882	0.6131	0.6029
20	0.5850	0.6067	0.6230	0.6144
19	0.5896	0.6219	0.6266	0.6200
18	0.6010	0.6322	0.6303	0.6259

Table 3: Average ρ scores for subsets of annotators.

In parallel, we calculate Fleiss' κ and the Jaccard index for the same subsets. These values can be seen in tables 4 and 5. Our motivation was to find the optimal compromise between a reasonable agreement (which of course increases with less annotators) and still keeping a sufficient number of annotators. The increase of our measures declines when going below 20 annotators. Therefore, we decide to continue working with this subset, which also ensures that a large amount of difficult terms is included: table 6 shows the number of terms judged higher than 2.5 on average, as relying on the upper median. The subset of 20 annotators seems to accommodate our imbalance of simple and difficult terms.

#Anno.	Cooking	DIY	Auto.	All
25	0.5006	0.4710	0.4348	0.4589
24	0.5214	0.4854	0.4393	0.4695
23	0.5325	0.4937	0.4420	0.4756
21	0.5412	0.5067	0.4514	0.4856
20	0.5446	0.5110	0.4553	0.4895
19	0.5439	0.5146	0.4539	0.4895
18	0.5517	0.5212	0.4572	0.4946

Table 4: Average Jaccard for subsets of annotators.

[2] The original instructions were given in German.

#.	Cooking			DIY			Automotive			All		
	4	3	2	4	3	2	4	3	2	4	3	2
25	0.2411	0.2889	0.3566	0.2397	0.3081	0.3396	0.2296	0.2765	0.3978	0.2408	0.2932	0.3890
24	0.2620	0.3160	0.3928	0.2606	0.3338	0.3589	0.2359	0.2852	0.4065	0.2544	0.3109	0.4066
23	0.2744	0.3284	0.4097	0.2707	0.3466	0.3616	0.2399	0.2908	0.4064	0.2621	0.3200	0.4113
21	0.2968	0.3509	0.4370	0.2940	0.3757	0.3871	0.2557	0.3092	0.4328	0.2810	0.3417	0.4371
20	0.3119	0.3688	0.4544	0.3054	0.3903	0.4043	0.2635	0.3185	0.4428	0.2910	0.3538	0.4498
19	0.3121	0.3642	0.4679	0.3124	0.3926	0.4343	0.2614	0.3136	0.4495	0.2915	0.3507	0.4627
18	0.3204	0.3751	0.4752	0.3191	0.4013	0.4426	0.2643	0.3172	0.4512	0.2962	0.3569	0.4667

Table 5: Average κ for different subsets of annotators with respect to the number of rating categories, i.e., across all 4 categories on our scale of difficulty and also for 3 classes (categories 1 vs. 2+3 vs. 4) and for 2 classes (1+2 vs. 3+4).

#Anno.	Cooking	DIY	Auto.	All
25	44	66	225	335
24	46	70	235	351
23	45	66	230	341
21	47	73	232	352
20	49	80	243	**372**
19	49	78	231	358
18	52	79	236	367

Table 6: Number of words with median rating $\geq$ 2.5.

7. Final Dataset

7.1. Selected Annotations

Table 7 shows the Fleiss' κ and the Spearman's ρ correlations for the 20 annotations where annotators agreed most. We can see that the results are rather low for Fleiss' κ; but Spearman's ρ –which measures the rankings rather than the actual values– is sufficiently high, with an overall correlation of 0.614.

domain	Spearman's ρ	Fleiss' κ
Cooking	0.585	0.312
DIY	0.607	0.305
Automotive	0.623	0.264
total	0.614	0.291

Table 7: Inter-annotator agreement for 20 annotators.

7.2. Mapping of Annotations to GS Classes

There are a number of options regarding which values actually constitute the gold annotation scores and classes for each term. We decide for five different gold standards as based on the annotation of our 20 raters, to enable various perspectives for interpretation.

(i) **Majority voting**: 4 classes.

- The GS class is the class which was chosen by the majority of the annotators (1, 2, 3 or 4).
- If two classes were chosen equally often, we take the higher class value.

(ii) **Median**: 4 classes.

- The GS class is the upper median, i.e., if the median value is .5 it is rounded up. Table 8 shows how often rounding up was performed.

(iii) **Majority**: 2 classes (binary: 1 vs. 2+3+4).

- Terms with the majority for class 1 are in one class, the rest (2, 3, 4) is in the other class.

(iv) **Median**: 2 classes (binary: 1+2 vs. 3+4).

- Terms with median < 2 are in one class, terms with median ≥ 2.5 are in the other class.

(v) **Mean**: Mean value of all ratings.

	1.5 → 2	2.5 → 3	3.5 → 4
Cooking	11	3	1
DIY	3	9	4
Automotive	18	16	19

Table 8: Number of median gold values rounded up.

8. Dataset Statistics

Table 9 shows the distribution of the ratings of the selected 20 annotators. We can see that 36.8% of the ratings are of class 1, and only 13.5% are of class 4; this reflects our intuition that the difficulty of a term is imbalanced across classes.

Class	Cooking	DIY	Auto.	All
1	2,274	1,925	3,381	7,580 (36.8%)
2	1,427	1,448	3,014	5,889 (28.6%)
3	811	945	2,598	4,354 (21.1%)
4	348	542	1,887	2,777 (13.5%)

Table 9: Number of ratings per class, with the total number of ratings = 20,600.

In Figure 2 all terms are sorted by their **mean** rating over all 20 annotators. The graph also visualizes the **standard deviation** of the values showing that only terms in class 1 have been rated with total agreement. With an increasing mean difficulty value we also have an increase of the standard deviation; this reflects the uncertainty of people when rating terms which are not obviously simple. With respect to terms with a very high mean difficulty value, raters again agree slightly more, as shown by a slight decrease of the standard deviation.

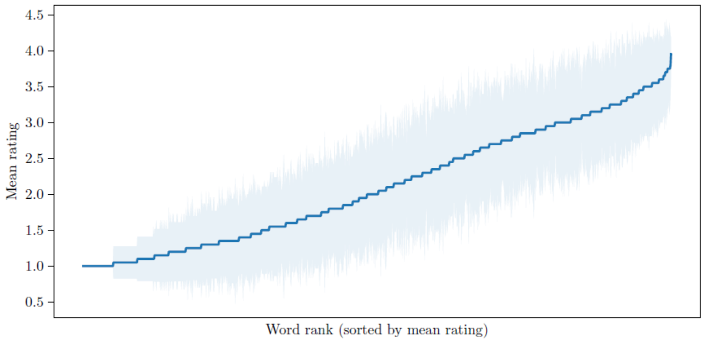

Figure 2: Mean ratings of all words in ascending order; includes standard deviation.

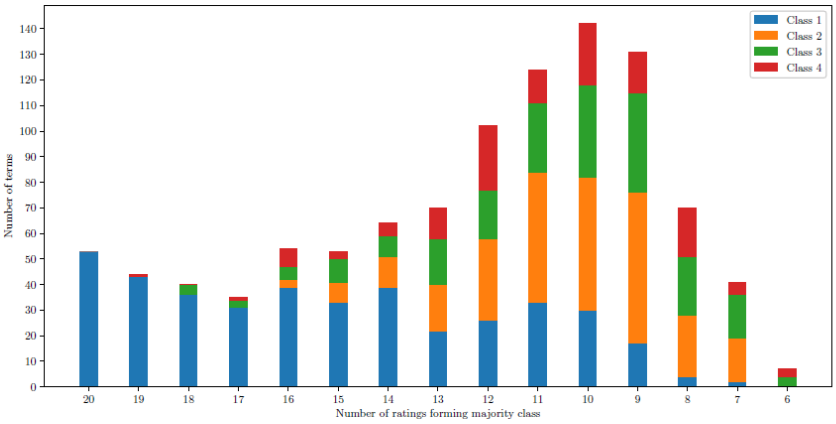

Figure 3: Number of ratings underlying the majority 4-class assignment.

When considering our **majority** gold values, we are interested in the strength of the agreement on the majority class of a term. Figure 3 shows the number of terms with respect to the number of ratings actually representing the majority class. For 53 terms there is a complete agreement across all 20 annotators on the rating of class 1 (see left-most bar). 19 annotators (second bar from the left) agreed on the rating of class 1 for 43 terms and on the rating of class 4 for one term. Overall, we can see that the terms where 14–20 annotators agreed on the class value were mostly from class 1 (blue parts of the bars). For classes 2–4 there is less agreement on the exact class of a term; for example, for most terms with ratings of class 2 (orange parts of the bars) we only found agreement across 7–13 annotators.

Figure 4 breaks down the absolute distribution across the majority classes for the individual domains, with a total of 408 terms in class 1, 276 terms in class 2, 212 terms in class 3 and 134 terms in class 4. For all three domains we observe an almost linear decrease of the number of terms with increasing difficulty (i.e., higher class value). Since these numbers are based on different absolute term numbers regarding cooking/DIY and automotive, Figure 5 illustrates the proportions of terms in one class with respect to each domain. Now it is even more obvious that the proportion of difficult terms is largest in the automotive domain, whereas it is the lowest in the cooking domain where most of the terms (≈51.85%) are in class 1.

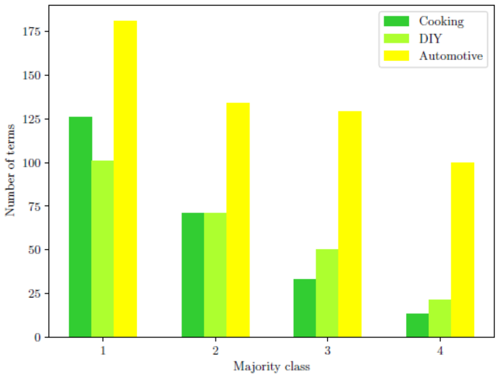

Figure 4: Number of terms per class (majority).

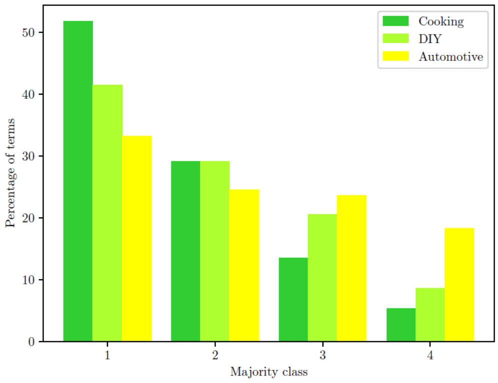

Figure 5: Proportions of terms per class (majority).

The absolute distribution of our binary majority classes (class 1 vs. classes 2–4) can be found in Figure 6 where we have 408 terms in class 1 versus 622 terms in the other three classes.

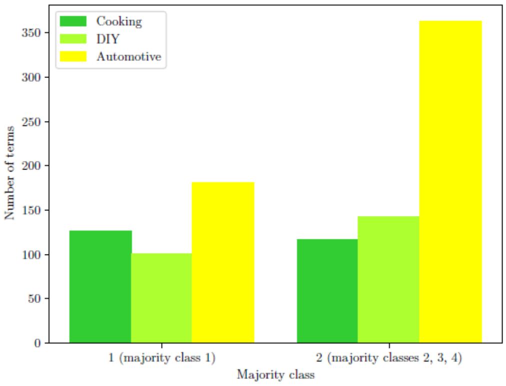

Figure 6: Distribution of binary classes (majority).

Figures 7–9 illustrate the same information for the **median** gold values as Figures 4–6 did for the majority values. Figure 7 in comparison to Figure 4 shows that relying on median values leads to a shift towards more higher ratings (i.e., more difficult terms) in total, but there are still a low number of ratings in class 4. This is also reflected in Figure 8. The binary version of median values where we chose classes 1 and 2 versus classes 3 and 4 as our binary options is depicted in Figure 9.

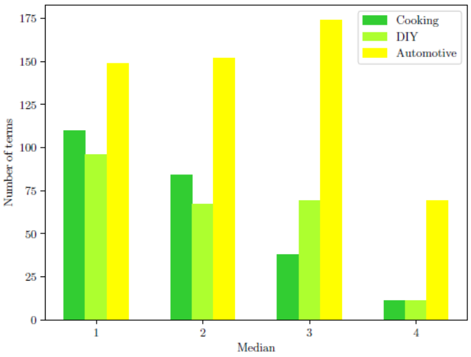

Figure 7: Number of terms per class (median).

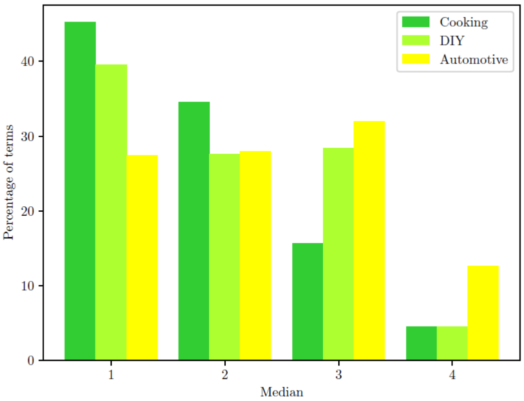

Figure 8: Proportion of terms per class (median).

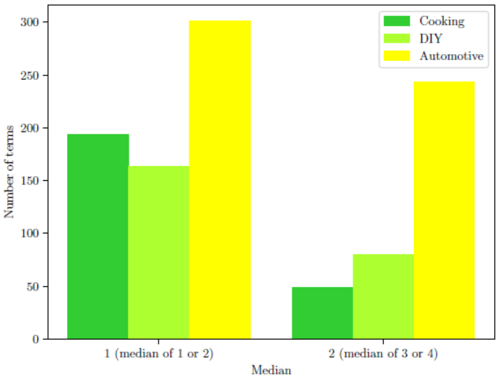

Figure 9: Distribution of binary classes (median).

Based on the fact that we decided to select automotive terms across different numbers of constituents (2–4 constituents), Figure 10 illustrates how the number of terms per median class correlate with the number of constituents. Figure 11 makes the three groups more easily comparable by focusing on the proportions. As expected intuitively, we can see that terms with more constituents are generally perceived as more difficult, with however the relative majority of four-part compounds in class 3.

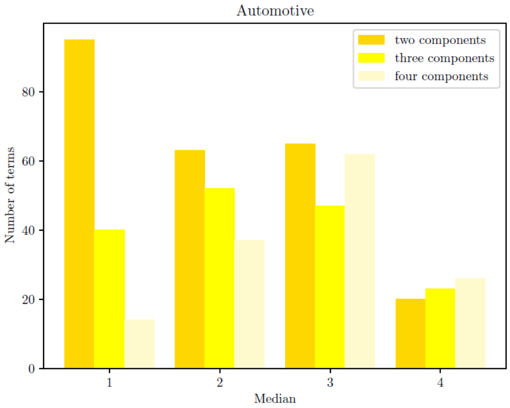

Figure 10: Median distribution of automotive terms across the number of components.

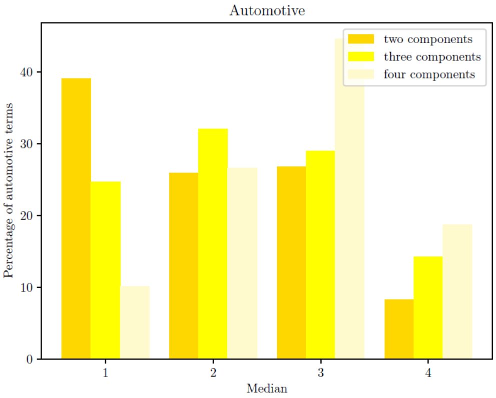

Figure 11: Proportions of classes across the number of components (median).

9. Discussion

Based on variants of gold standards for term difficulty of compounds and by relying on fine-grained analyses of the annotators' annotations and (dis)agreements we can summarise a number of observations for that and why it remains difficult to annotate term difficulty with strong agreement, and what our recommendations are for future collections of difficulty ratings.

Standard deviations of mean difficulty values indicated that most agreement among annotators is achieved for the extremes, i.e., for clearly easy and for some clearly difficult compounds. This is in accordance with a general tendendy in semantic variable ratings that has been observed before (Pollock, 2018).

An analysis of how many annotators agreed on the individual compounds' difficulty ratings confirmed that the vast majority of compounds where most of the annotators agreed is from class 1 (i.e., the easiest compounds) and that less agreement is achieved for classes 2–4. This leads to the conclusion that annotators agree on easy vs. difficult domain-specific compounds but that a more fine-grained distinction of difficulty is not meaningful. For a gold standard of difficulty we thus suggest to employ binary rather than more fine-grained decisions.

It remains as a core question what actually makes a term difficult, and why some terms are perceived as easier than others. Looking into how the number of constituents of a compound influenced the annotators' ratings on difficulty, we observed that the more complex a domain-specific compound is, the more difficult it appears to the annotators. We could also see that the proportions of difficult domain-specific terms are larger for the automotive in comparison to the cooking and the DIY domains, where the latter are considered as more related to everyday experience than the former.

But a lot of questions for future explorations remain. Which further factors play a role in domain-specific term difficulty? Is difficulty mainly due to less common usage, i.e., frequency or productivity? Is this the reason why more complex terms are more difficult and why cooking terms are easier than automotive terms? And what is the role of the constituents, e.g., is a compound term already considered easy if only the head is known, even if the exact definition of the term remains unclear? For example, would an annotator rate the compound term *Kärnersbraten* ("Kärner's Roast") as easy while only knowing that it denotes some kind of roast and that *Kärner* is a proper name, but without being aware of the exact definition?

Figure 12 takes a first step into addressing these questions and looks into the role of compound and constituent frequencies and productivities when judging the compounds' degrees of difficulty: Relying on the majority binary ratings (1 vs. 2+3+4) for the automotive domain we plotted the difficulty classes for the 2 × 81 most extreme[3] compounds regarding the respective empirical properties in both the general-language corpus (left panel) and the domain-specific corpus (right panel). For example, the first line of plots shows the proportion of easy compounds (class 1) and difficult compounds (classes 2–4) for the 81 most low-frequent compounds (blue bars) and the 81 most high-frequent compounds (orange bars). Across the ten plots we can see that annotators perceived compounds with high frequencies and compounds with high-frequency and high-productivity modifiers in the general-language corpus as easier than the respective low-frequency/-productivity compound sets. The influence of head frequency and head productivity in the general-language corpus regarding the difficulty ratings was less strong, and ditto for the influence of most empirical properties in the domain-specific corpus. So overall the general-language frequencies and productivities of compounds and constituents played a crucial role in how difficult the automotive terms appeared to the annotators. For the cooking and the DIY domains the insights are similar; however, in the cooking domain the modifier frequencies and productivities in the domain-specific corpus provided a stronger influence on the judgments; even more so for the DIY domain, where in addition also the head properties played a stronger role.

Finally, any annotation guidelines requiring an annotator to consider the difficulty of a term independently of one's own knowledge is hard and very subjective. One has to reflect if a word is actually an easy term or whether it just appears to be easy because one has some kind of expert knowledge or is to some degree familiar with the domain. We tried to counteract this problem by asking the annotators to be as objective as possible but annotations will, of course, always remain subjective to some extent.

10. Conclusion

This study described the creation of a noun compound difficulty dataset for German closed compounds. We focused on domain-specific compounds occurring in the domains cooking, DIY and automotive. Compounds were selected from domain-specific corpora by using a compound splitter. Then a filtered and balanced subset of compounds was annotated for difficulty. A quantitative dataset analysis was conducted following the annotation process, and we found that annotators agree on easy vs. difficult domain-specific compounds but that a more fine-grained distinction of difficulty is not meaningful. Furthermore, looking into compound and constituent frequencies and productivities revealed that the empirical properties play an important role in the perception of compound term difficulty.

Our novel dataset of difficulty ratings for German closed noun compounds is publicly available from `https://www.ims.uni-stuttgart.de/data/term-compound-difficulty`.

11. Acknowledgements

The research was supported by Robert Bosch GmbH with PhD/Master research grants for Anna Hätty and Julia Bettinger, respectively. We also thank Jonas Kuhn from the University of Stuttgart for funding the annotation study.

[3] I.e., we compared the two extreme thirds of 81 compounds each and ignored the middle third of 81 compounds.

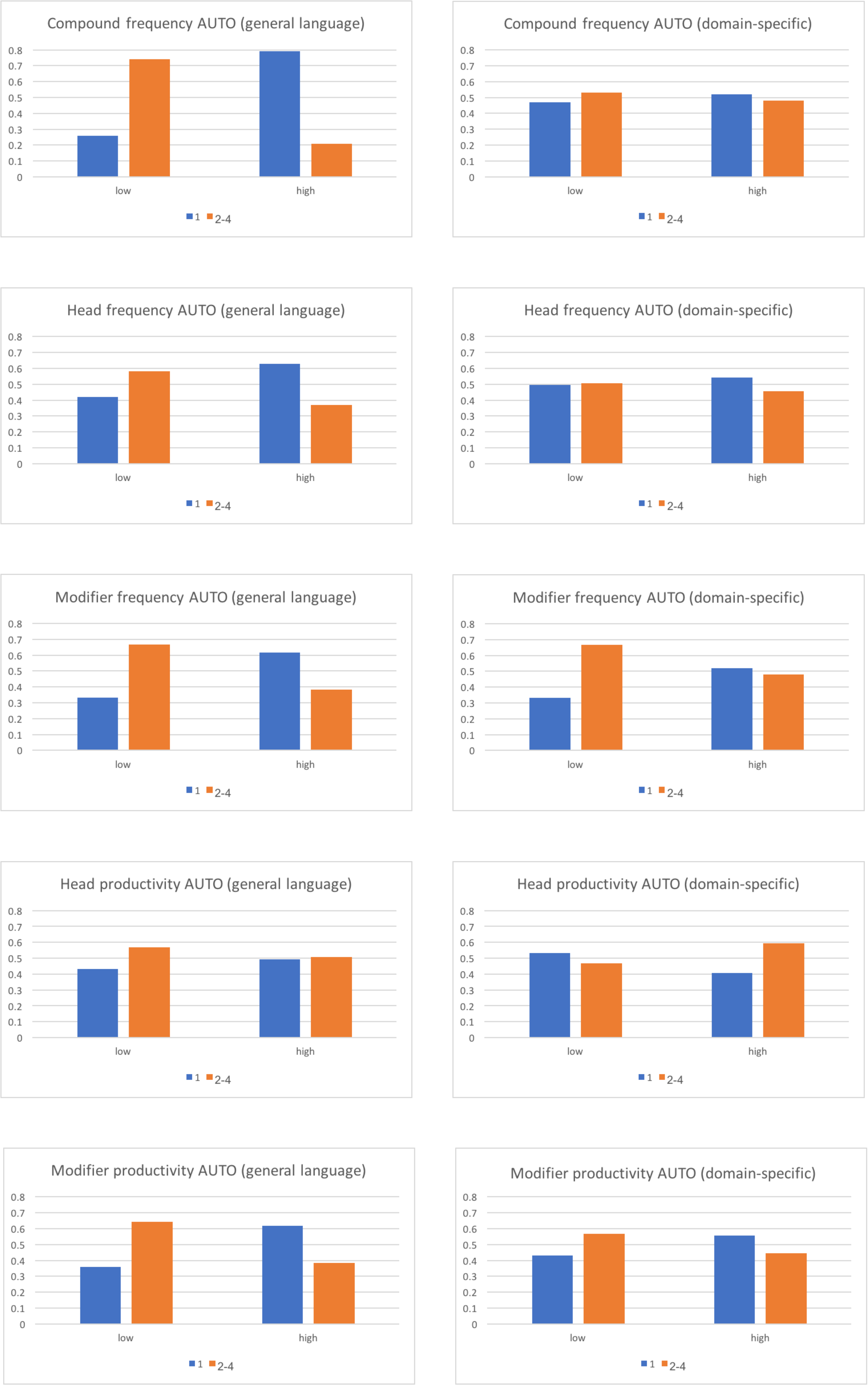

Figure 12: Proportions of majority binary classes with respect to compound and constituency properties.

12. Bibliographical References

Bouamor, D., Llanos, L. C., Ligozat, A.-L., Rosset, S., and Zweigenbaum, P. (2016). Transfer-based learning-to-rank assessment of medical term technicality. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Paris, France. European Language Resources Association.

Clouet, E. L. and Daille, B. (2014). Splitting of compound terms in non-prototypical compounding languages. In *Proceedings of the First Workshop on Computational Approaches to Compound Analysis*, pages 11–19, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Elhadad, N. (2006). Comprehending technical texts: Predicting and defining unfamiliar terms. In *AMIA Annual Symposium Proceedings*, volume 2006, page 239. American Medical Informatics Association.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Grabar, N., Hamon, T., and Amiot, D. (2014). Automatic diagnosis of understanding of medical words. In *Proceedings of the Third Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 11–20, Gothenburg, Sweden. Association for Computational Linguistics.

Jaccard, P. (1902). Lois de distribution florale dans la zone alpine. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 38:69–130.

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Kandula, S., Curtis, D., and Zeng-Treitler, Q. (2010). A semantic and syntactic text simplification tool for health content. In *AMIA Annual Symposium Proceedings*, volume 2010, page 366. American Medical Informatics Association.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*.

Pollock, L. (2018). Statistical and methodological problems with concreteness and other semantic variables: A list memory experiment case study. *Behavior Research Methods*, 50:1198–1216.

Siegel, S. and Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Boston, MA.

Vydiswaran, V. V., Mei, Q., Hanauer, D. A., and Zheng, K. (2014). Mining consumer health vocabulary from community-generated text. In *AMIA Annual Symposium Proceedings*, volume 2014, pages 1150–1159. American Medical Informatics Association.

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

Zeng, Q., Tse, T., Divita, G., Keselman, A., Crowell, J., Browne, A., Goryachev, S., and Ngo, L. (2007). Term identification methods for consumer health vocabulary development. *Journal of Medical Internet Research*, 9(1).

Zeng-Treitler, Q., Goryachev, S., Tse, T., Keselman, A., and Boxwala, A. (2008). Estimating consumer familiarity with health terminology: A context-based approach. *Journal of the American Medical Informatics Association*, 15(3):349–356.