From term extraction to lemma selection for an electronic LSP-dictionary in the field of mathematics

Theresa Kruse[1], **Ulrich Heid**[1]

[1]Institute for Information Science and Natural Language Processing (IwiSt), University of Hildesheim, Universitätsplatz 1, 31141 Hildesheim, Germany
E-mail: theresa.kruse@uni-hildesheim.de, ulrich.heid@uni-hildesheim.de

Abstract

We work on term extraction for a corpus-based LSP-dictionary. Our field of study is the mathematical domain of graph theory. Our working hypothesis is that mathematics lends itself to a specific approach for term and information extraction with a lexicographical purpose. We compare different methods for term extraction: The first one combines pattern-based and statistical means implemented by Schäfer et al. (2015), the second one has been developed especially for mathematical texts using domain-specific definition patterns based on work in the tradition of Meyer (2001). Further comparisons are made with a list of term candidates which are not part of the general language lexicon used in a version of TreeTagger trained on news text (Schmid, 1994) and with the term extraction provided by Sketch Engine (Kilgarriff et al., 2014). We use manual annotation by three expert raters and inter-rater agreement with κ-statistics to compare and evaluate the approaches. Additionally, we qualitatively analyse the extracted results. For selecting the lemmas, we work with a German corpus of lecture notes, textbooks and papers.

Keywords: LSP-dictionaries; mathematics; pattern-based extraction; automatic creation; semantic relation

1. Introduction

Our work on term extraction and lemma selection evolved as part of a project on creating an online LSP-dictionary[1] covering the domain of graph theory, a part of mathematics. Its target group are students. The dictionary is based on scientific and didactic literature from the domain: textbooks, course material and specialised publications. It is intended to cover the central terminology of graph theory as well as items from other mathematical domains which are needed to understand graph-theoretical literature. The dictionary will have an ontology as its backbone and will give equivalents in German and English, as well as definitions and semantically related terms. Most of these relations correspond to lexical semantic relations known from linguistics, such as hyperonymy, only some relations are domain-specific.

Our working hypothesis is that we can rely exclusively on (definitional) patterns to extract terms from the graph-theoretical texts and that we do not need statistical approaches, because mathematical texts contain highly standardised definitions.

In Section 2, we give a short overview of methods for term extraction. We compare different methods for term extraction to investigate our hypothesis: A rather traditional pattern-based one combined with statistics as described by Schäfer et al. (2015) and one that only relies on domain-specific patterns in the tradition of Meyer (2001). We extract a list of term candidates with these tools and add items from the corpus which are not part of the general language lexicon used in a version of TreeTagger trained on news text (Schmid, 1994).

Three expert raters decided in two rounds which of these candidates should become lemmas of the dictionary. We present the results of this rating in Section 3. As a consensus on refined guidelines for lemma selection preceded the second selection round, we also use

[1] LSP stands for Language for Special Purposes.

the resulting lemma list to evaluate the contribution of each term extraction method to the creation of the lemma list, i.e. the results of the rating are used to evaluate the different methods in Section 4. We are aware of the methodological problem that lies in the bias towards the tool output, because we may systematically miss lemma candidates not found by any of the approaches.

A further comparison is made with the term extraction provided by *Sketch Engine* (Kilgarriff et al., 2014). Section 5 brings together the results of the evaluations. We conclude in Section 6.

2. Related work on term extraction

Different approaches to term extraction appeared over the last 20 years: Cabre & Vivaldi Palatresi (2013) give an overview of the state of the art of around 2010 and distinguish linguistic, statistical and hybrid methods. An overview of current experiments based on Machine Learning (ML) can be found e.g. in Hätty (2020), while Hätty herself combines different traditional as well as ML approaches.

Cabre & Vivaldi Palatresi (2013) name three criteria for terms: unithood, termhood and specialised usage. Unithood and termhood are also common benchmarks in evaluating the results of automated term extraction (cf. Zadeh & Handschuh, 2014). *Termhood* is the extent to which a candidate is actually a term. *Unithood* is a measure of the association between different components of a multiword term candidate and is thus similar to some measures of collocational strength.

Extraction tools have to find single word terms (SWT) as well as multiword terms (MWT). Cabre & Vivaldi Palatresi (2013) indicate frequency counts, frequency comparison and pattern search as the main methods for extracting SWT and linguistically based pattern search, keyword-in-context and statistical techniques for MWT. All methods may be combined.

Frequency comparisons include contrastive approaches in which the frequency of each candidate in the specialised text is compared to a reference corpus from general language. Several measurements exist for such a comparison: e.g. frequency profiling (Rayson & Garside, 2000), the C-NC value (Bonin et al., 2010) or the modified weirdness measure (Kochetkova, 2015). *Sketch Engine* (Kilgarriff et al., 2014) also uses a frequency comparison technique.

Often, a scoring or ranking of the results follows the extraction itself. Depending on the domain, terminologies may exist as a reference for evaluation. Especially when working on variants this constitutes a useful approach (cf. Zadeh & Handschuh, 2014). Bernier-Colborne (2012) introduce a method for creating a gold standard from a corpus which may also be used for these purposes.

Recent automatic term extraction uses Machine Learning. Different approaches have been developed in recent years (Rigouts Terryn et al., 2020). We give some examples in the following.

Dobrov & Loukachevitch (2011) combine frequencies from domain-specific texts and search engines with a domain-specific thesaurus. Conrado et al. (2013) combine multiple

features like term frequency, part of speech and context for their ML-based extraction. Fedorenko et al. (2014) compare term extraction based on ML using different features with voting algorithms and conclude that the ML-methods outperform the others.

It has been shown that word embeddings are also helpful for term extraction. Amjadian et al. (2016) use distributed vectors based on the regression model GloVe (Pennington et al., 2014), which constitutes a step towards language independent term extraction and combines linguistic and statistical approaches. They also evaluate their method on mathematical texts, namely five English high school textbooks. They do not indicate any difficulties that would be due to the domain. Their distributed vectors work best as a filter and not directly applied to a corpus (Amjadian et al., 2018).

Wang et al. (2016) also use word embeddings with a focus on reducing the amount of labelled data. Therefore, they use co-training (Blum & Mitchell, 1998): First, only a part of the data is labelled and the most probable labels are taken into consideration. The tool works iteratively this way.

Some term extraction tools were especially developed for lexicographic purposes, such as the *Sketch Engine* term extraction or the procedure used by Heid & Weller (2010) based on dependency parsing to extract MWT. *Sketch Engine* (Kilgarriff et al., 2014; Jakubíček et al., 2014) annotates with the RFTagger (Schmid & Laws, 2008) for a pattern-based term extraction. We use this tool on our data in Section 5.4 to have another comparison. Pollak et al. (2019) present a different approach for lexicography which combines frequency methods with word embeddings.

One of the tools tested in our lemma selection experiments is the term extractor implemented by Schäfer et al. (2015) in line with the traditional hybrid approach (cf. also Roesiger et al., 2016). Schäfer et al. (2015) focus on adjectives and nouns and implement three steps: First, they select nominal candidates by part-of-speech tagging; secondly, they take the syntactic validity of noun phrases into account and thirdly, they use statistical measures. They extract the following POS-patterns based on regular expressions, where N is the POS tag noun, *Adj* adjective, P preposition, *Adv* adverb and D determiner:

- (Adv? Adj? Adj)? N
- (N D)? (Adv? Adj)? N P D? (Adv? Adj)? N
- (Adv? Adj)? N D (Adv? Adj)? $N_{genitive}$

For removing noise they use the *c*-value score (Frantzi & Ananiadou, 1996) and combine constituency and dependency parsing (Bohnet, 2010; Choi et al., 2015; Roesiger et al., 2016). The *c*-value is an established domain-independent (Frantzi et al., 2000) measure for ranking extracted terms based on frequency and on the usage of an item in MWTs. Schäfer et al. (2015) evaluate their tool on texts from the domain of do-it-yourself projects and get an F-score of 0.59 with a precision of 0.48 and a recall of 0.77. We present our results with this tool in Section 4.1.

3. Extracting and categorizing the lemmas

3.1 Expert raters

We work with a corpus of German lecture notes, textbooks and papers from the mathematical sub-domain of graph theory. It contains 882,910 tokens with 31,106 types.

We extract a list of 4205 lemma candidates from it and give it to three expert raters. Section 4 describes the process of selecting the terms in the list.

The classification consists of two steps: First, the raters work individually and independently. We analyse their results and make out systematic differences and disagreement concerning lexicographic and linguistic aspects. In a subsequent adjudication step, the raters discuss the (types of) phenomena which led to their divergent classifications. Finally, we ask them to agree on common guidelines for these cases.

The three expert raters come from different backgrounds in graph theory. All of them have studied mathematics, have didactic experience in mathematics and work with academic graph theory from different perspectives. In the first selection round, we simply ask them to decide for each candidate whether it should be given lemma status in the planned dictionary: „Bitte beantworten Sie für jeden Begriff in der Liste die folgende Frage: Soll es im geplanten elektronischen Wörterbuch einen Eintrag zu diesem Lemma geben?“[2]. We also ask them to propose further terms and to comment on their choices in cases of uncertainty.

All raters are familiar with the idea of the project to create an electronic dictionary for the domain of graph theory which can be used by students. One of the raters is aware of the semantic category system which is used on the lemma list at a later point in the lexicographic process.

	individual classification		after discussion	
	number of terms	percentage	number of terms	percentage
3 votes	383	9.11 %	1077	25.64 %
2 votes	783	18.62 %	376	8.94 %
1 vote	897	21.33 %	334	7.94 %
0 votes	2142	50.94 %	2417	57.48 %

Table 1: Results of the expert raters

Table 1 shows the results of the individual classification. The raters consider only about half of the extracted items as useful for the dictionary. In the later sections we investigate the reasons for the low quality of the extraction tools.

We calculate the inter-rater agreement with κ-statistics (Fleiss, 1971) and get $\kappa = 0.3484$. The agreement within the categories is $\kappa_{\text{in}} = 0.3489$ and $\kappa_{\text{out}} = 0.3479$. A pairwise comparison between the raters is provided in Table 2. The agreement in this first round is only *fair* or at most *moderate*, in terms of the terminology proposed by Landis & Koch (1977). This result confirms the observation made by Hätty (2020) that intuitive notions of termhood vary considerately between individual raters; this also seems to be the case with experts from the same domain.

[2] Engl.: For each term in the list, please answer the following question: Should there be an entry for this lemma in the planned electronic dictionary?

		Rater 1		
		in	out	
Rater 2	in	0.2499	0.0404	0.2903
	out	0.1960	0.5137	0.7097
		0.4459	0.5541	

$\kappa = 0.5047$

		Rater 2		
		in	out	
Rater 3	in	0.0968	0.0259	0.1227
	out	0.1936	0.6837	0.8773
		0.2904	0.7096	

$\kappa = 0.3578$

		Rater 1		
		in	out	
Rater 3	in	0.1127	0.0100	0.1227
	out	0.3332	0.5441	0.8773
		0.4459	0.5541	

$\kappa = 0.2526$

Table 2: Agreement between raters

We subsequently initiate the adjudication discussion mentioned above to understand the raters' reasoning underlying their decisions, and to jointly develop refined guidelines for lemma selection.

Among other case-by-case decisions, the following aspects seem to be crucial reasons for different classifications: First, the degree to which the translation between German and English is considered as difficult for the intended public of the dictionary. We work on a bilingual dictionary containing equivalents as well as onomasiological and definitional information on the terms. The annotators have a different focus on these aspects and therefore terms like *Satz von Petersen* (Engl. *Petersen's theorem*) are excluded by one rater because the students should have no difficulties in translating them. A similar reasoning holds for certain compound terms. In the discussion, the raters decide to include these terms as they belong to a given conceptual category in the final dictionary (Kruse & Heid, 2020).

A second difficulty is common mathematical terminology which is not particularly typical for the sub-domain of graph theory, such as terms referring to set theory. This issue is an instance of the more general problem of the delimitation of (sub-)domains in terminology, as addressed e.g. in the model of Roelcke (2010) of intra-subject vs. inter-subject terminology (*intrafachlicher* vs. *interfachlicher Fachwortschatz*). Some annotators include these terms because they are basic for anyone learning graph theory, and others exclude them because they are not specific of the sub-domain. Hence, we add a category for these general terms to our classification system (cf. Section 3.2).

The third main aspect that leads to differences among the raters are term variants. We already gave an overview on variants of our domain in Kruse & Giacomini (2019). Two annotators decide to only include one (primary) variant into the lemma list of the dictionary. After the discussion, they include all variants into the lemma list. Possibly, some of them will appear in the dictionary as cross-reference entries, i.e. as links to another variant.

Another issue is the handling of mathematical symbols. The raters decide to exclude them because the symbols need a verbalisation which requires some further, possibly very specific, lexicographic devices.

The raters decide to include compounds of variables and words like *2-regulär* only with the most common abbreviation, like *k-regulär*. Only few exceptions are made for terms which have a special significance in graph theory, e.g. *2-dimensional*. We cannot treat these cases like variants because on a semantic level they are at most hypernyms. For example, *2-regulär* is a special case of *k-regulär* and some properties are valid for only certain values of k. Therefore, they cannot be treated on the same level.

Another discussion point are combinations of terms with words from general language like *Anzahl an...* (Engl. *number of...*). In these cases, the raters decide to only include the terminological parts as long as the added word is terminologically irrelevant. Otherwise, obviously the whole term is included, as is the case with *Kuratowskimenge* (Engl. *Kuratowski set*).

Further, the raters discuss which combinations are considered as a MWT. One example are combinations with *maximal* and *minimal*. Mostly, these combinations are not terminologically relevant, but there are specific exceptions, e.g. *maximal Matching* which is a lot more used than *minimal Matching*. Thus, these decisions are made on a case-by-case basis.

It is also very common in mathematics to have negated compounds with *nicht-* (Engl. *not-*) and *-frei* (Engl. *-free*). If one knows the other part, they are self-explanatory and therefore not included in the dictionary, but their positive counterparts will be.

After re-annotating the data and taking the results of the discussion into account we get $\kappa = 0.7500$. Table 1 gives the results of this second step and Table 3 the pairwise comparison between the raters. We include the candidates with at least two votes in the dictionary, and thus our final lemma list contains 1,453 lemmas.

Thus, overall, the adjudication process was also a process of refining the lexicographic lemma selection principles, and it was massively dependent on the peculiarities of the domain and on the specialised vocabulary to be dealt with, but also on decisions concerning a homogeneous lexicographic treatment of certain classes of items. Nevertheless, we have to admit that the selection remains partly random because the raters' prompt does not give clear criteria and can be individually interpreted, as the discussion has shown. It might be useful to use these criteria for another annotation with new raters to get more generalisable results.

		Rater 1		
		in	out	
Rater 3	in	0.1085	0.0040	0.1173
	out	0.2627	0.6200	0.8827
		0.3712	0.6288	

$\kappa = 0.7145$

		Rater 1		
		in	out	
Rater 2	in	0.2133	0.0410	0.2543
	out	0.1579	0.5878	0.7457
		0.3712	0.6288	

$\kappa = 0.7909$

		Rater 2		
		in	out	
Rater 3	in	0.0843	0.0330	0.1173
	out	0.170	0.7127	0.8827
		0.2543	0.7457	

$\kappa = 0.7476$

Table 3: Rater results after discussion

3.2 Categorisation

We manually assign the chosen terms to the following categories: ALGORITHM, MAPPING, PART (of a graph), PERSON, PROBLEM, THEOREM, TYPE (of a graph), PROPERTY (of a graph), ACTIVITY and GENERAL. Kruse & Heid (2020) provide a detailed description of these categories, except for GENERAL which is the category mentioned above containing all the general mathematical terms which are a prerequisite to but no direct part of graph theory. In the final dictionary the category of each item defines the microstructure of its entry.

Table 4 shows the distribution of the 1,453 lemmas over the ten categories. Almost a third of the lemmas belongs to the category PART (of a graph), followed by PROPERTY (of a graph) and TYPE (of a graph). These three constitute the majority of the concepts used in graph theory and in mathematics in general, as one has certain objects (PARTS and TYPES) for which PROPERTIES are defined.

4. Term extraction

In the following, we present our methods for term extraction. One has to keep in mind that our results are biased because the raters could only decide upon the extracted terms, not on an independent list. Nevertheless, they had the opportunity to add terms to the list on their own. We choose this workflow because there were no capacities for our raters to annotate the whole corpus of almost 900,000 tokens for establishing an independent gold standard.

Category	Number	Percentage
Part	411	28.29 %
Property	263	18.10 %
Type	162	11.15 %
General	153	10.53 %
Theorem	146	10.05 %
Mapping	128	8.81 %
Person	89	6.13 %
Algorithm	58	3.99 %
Problem	35	2.24 %
Activity	8	0.55 %

Table 4: Distribution of lemmas over categories

4.1 Combination of frequencies and patterns

We extract 2,416 potential lemmas with the method by Schäfer et al. (2015). In the following, we refer to this method as the T-method. We remove candidates from the list which result from noise in the corpus data, e.g. because of formatting fragments of formulas like *IJI-IJI*. 2,229 (92.26 %) lemma candidates remain. Only then did the raters receive the list. For precision and recall we calculate with this figure.

We use the 1,453 lemmas retained in the selection process from Section 3 as a gold standard for calculating precision p, recall r and F-score F. We can do that because we asked the raters to name further terms which they would like to include into the dictionary, and they did not give any. 643 candidates in the T-list got a vote by at least two raters.

$p_T = \frac{643}{2229} = 0.2885, r_T = \frac{643}{1454} = 0.4422, F_T = 0.3492$

In their paper Schäfer et al. (2015) get $p = 0.48$, $r = 0.77$ and $F = 0.59$, which is higher than in our experiment. Nevertheless, their data is not completely comparable with ours, because our data contains lemmas which might be terminological for mathematics but not in our specific sub-domain of graph theory.

4.2 Domain-specific patterns

The second extraction method is based on the hypothesis that we do not need any frequency measurements for term extraction in mathematics because the language is highly structured. Thus, we solely use domain-specific patterns. We call this method the P-method and identify the following words as pattern indicators: *bestehen aus*, *bezeichnen, definieren, erklären, haben, heißen, sein, Name, nennen, sagen, schreiben, sprechen, verstehen*[3]. The P-method returns 3,071 lemma candidates.

We carry out the same adjustments as described in Section 4.1 before we give the list to the raters. 1,797 (58.52 %) of the candidates remain after the adjustments. The raters

[3] Engl.: *consist of*, *denote*, *define*, *explain*, *have*, *be called*, *be*, *name*, *called*, *say*, *write*, *speak of*, *understand*

give 506 of the remaining terms at least two votes. This percentage of potential useful lemmas is lower than what we got with the T-method. This is maybe due to the fact that we did not include any measures of frequency. We get the following results for precision, recall and F-score:

$p_P = \frac{506}{3072} = 0.1647, r_P = \frac{506}{1454} = 0.3480, F_P = 0.2236$

These values are also lower than those of the T-method. There are some possible reasons for that which we examine in Section 5.

4.3 Comparison with unknowns

The candidate list for the raters combines the terms extracted by the two methods described in the previous sections. The list is supplemented with data generated during the correction process of the corpus. It contains words which were labeled as unknown by the TreeTagger (Schmid, 1994), trained on general language data (news text). We refer to this list as the U-list; and it contains 1478 potential terms. As the tagger operates on single word forms, only SWT appear on the list, including compounds like *(k+1)-elementig* (Engl. *(k+1)-element*) or *nicht-planar* (Engl. *non-planar*). We also calculate precision, recall and F-score to compare with the other methods.

$p_U = \frac{830}{1478} = 0.5616, r_U = \frac{830}{1454} = 0.5708, F_U = 0.5662$

These values are much higher than those obtained with the other methods because the U-list is not produced by means of data extraction, but through manual additions to the tagger lexicon. Thus, it can only be regarded as a sort of baseline with the downside that it does not contain any graph-theoretical terms that are polysemous with general language words (e.g. *Kante*, Engl. *edge* or *Ecke*, Engl. *node*).

5. Comparison of different methods

We see that the T-method produces less noise than the P-method because the T-method also includes a frequency measure whereas the other one does not. A pattern-based method works best on absolutely clean data, but formulas and abbreviations in mathematical texts lead to noise. Our corpus consists of sources with different formatting and file types, and we did not have the workforce to establish the same formatting for all texts. This has to be considered when working with mathematical texts, especially when they are combined from different sources.

The U-list has the best values, but here only SWT were included, and a lot of noise has been removed beforehand. Therefore, it can only serve as a reference. When using it for the lemma selection, it might be useful to include frequency figures and to only take lemmas with at least two mentions into consideration to improve the results of the P-method.

5.1 Comparison based on frequency

The Jaccard index J is a measure to determine how similar certain sets are (Jaccard, 1902). It is defined the following way for a number of n sets $A_1, ..., A_n$:

$$J(A_1, ..., A_n) := \frac{|A_1 \cap ... \cap A_n|}{|A_1 \cup ... \cup A_n|}$$

The Jaccard index takes values between $J = 0$ (if and only if $A_1 \cap ... \cap A_n = \emptyset$) and $J = 1$ (if and only if $A_1 = ... = A_n$). We have three sets: In P are the terms extracted by the pattern-based method, T gives the extracted terms with the tool by Schäfer et al. (2015) and U comprises the list of unknown words based on the lexicon by Schmid (1994). First, we take into account all the terms extracted:

$$\begin{array}{llll} |P| = 3071 & |P \cap T| = 596 & |P \cup T| = 4712 & J(P,T) = 0.1265 \\ |T| = 2237 & |P \cap U| = 240 & |P \cup U| = 4308 & J(P,U) = 0.0557 \\ |U| = 1477 & |T \cap U| = 262 & |T \cup U| = 3452 & J(T,U) = 0.0759 \\ & |P \cap T \cap U| = 113 & |P \cup T \cup U| = 5800 & J(P,T,U) = 0.0195 \end{array}$$

As these values are based on noisy data, it is preferable to compare only the terms which were finally chosen for the dictionary. However, still the values show no particularly high agreement between the sets:

$$\begin{array}{llll} |P_s| = 506 & |P_s \cap T_s| = 234 & |P_s \cup T_s| = 913 & J(P_s,T_s) = 0.2563 \\ |T_s| = 641 & |P_s \cap U_s| = 178 & |P_s \cup U_s| = 1157 & J(P_s,U_s) = 0.1538 \\ |U_s| = 829 & |T_s \cap U_s| = 200 & |T_s \cup U_s| = 1270 & J(T_s,U_s) = 0.1575 \\ & |P_s \cap T_s \cap U_s| = 89 & |P_s \cup T_s \cup U_s| = 1453 & J(P_s,T_s,U_s) = 0.0613 \end{array}$$

Another interesting set are those terms which are chosen for the final dictionary but only extracted by one of the tools. This affects 139 terms selected by the P-method, 193 terms from the T-method and 452 from the U-list.

5.2 Comparison based on categories

In Section 3.2 we divided the chosen lemma candidates into different categories. Now, we investigate how these categories are distributed among the terms depending on the extraction method. Most of the categories are evenly distributed over the different methods (cf. Table 5). The number of THEOREMS extracted by the P-method is so low because names of THEOREMS usually cannot be found with patterns as they are not part of definitions. The same applies for PERSONS. The number of ACTIVITIES extracted by the T-method is so high because it concerns nominalisations of verbs.

5.3 Error analysis

An error analysis in terms of classes of term candidates not found by the P- or the T-method is hard to realize, since almost no patterns emerge from these data. Nevertheless, some superficial remarks are possible: The P-method extracts some adjective-noun combinations, e.g. a few with the adjective *orientiert* (Engl. *oriented*). But something similar holds for the T-method, too: There are several combinations with *aufspannend*, *disjunkt* and *binär*, *hamiltonsch*, *eulersch*, *maximal*, *minimal*, *(stark) zusammenhängend*, *trennend*, *vollständig*[4]. All of them are combinations which appear in the texts with a certain frequency but are not part of the definitions on which the P-method mainly focuses.

[4] Engl. *spanning, disjoint, binary, Hamiltonian, Eulerian, maximum, minimum, (strongly) connected, separating, complete*

	Total		P-method		T-method		U-list	
PART	411	28.29 %	171	33.79 %	226	35.26 %	220	26.54 %
PROPERTY	263	18.10 %	105	20.75 %	32	4.99 %	187	22.56 %
TYPE	162	11.15 %	80	15.81 %	93	14.51 %	64	7.72 %
GENERAL	153	10.53 %	56	11.07 %	69	10.76 %	66	7.96 %
THEOREM	146	10.05 %	16	3.16 %	48	7.49 %	118	14.23 %
MAPPING	128	8.81 %	47	9.29 %	67	10.45 %	82	9.89 %
PERSON	89	6.13 %	12	2.37 %	67	10.45 %	21	2.53 %
ALGORITHM	58	3.99 %	10	1.98 %	20	3.12 %	41	4.95 %
PROBLEM	35	2.41 %	7	1.38 %	17	2.65 %	24	2.90 %
ACTIVITY	8	0.55 %	2	0.40 %	2	0.31 %	6	0.72 %
Σ	1453		506		641		829	

Table 5: Distribution over categories depending on extraction method

The P-method and the T-method miss out systematically on MWT when they show up in a context such as *NN heißt ADJ wenn*[5], i.e. in a non-adjacent form that fills the 'slots' of definition phrases. Thus, the P-method extracts the individual words but not their combination. This issue is an instance of the well-known problem of distinguishing clearly between SWT and MWT, and between MWT and collocations of SWT. As mentioned, the U-list does not contain MWT.

The U-list contains several unique terms not found by the other methods, e.g. combinations of a number and a word, like *3-regulär*. Such items cannot be found by the P-method, because definitions will only contain their generalised form, i.e. *k-regulär*. As different values are possible for k, low frequencies of the individual instances may also prevent the T-method from extracting words. The U-list also contains many compound nouns, e.g. with the heads *Kante* (Engl. *edge*), *Graph* (Engl. *graph*), *Ecke* (Engl. *node*), which are unknown to the tagger lexicon.

In the P-list we find two further classes of noise: Combinations of only two uppercase letters like *G N* and combinations of a nominal term and a single capital letter like *Graph G*. As they are excluded from the final lemma list we remove these 550 candidates. Such items do not appear in the results of the other two methods. With this modification, we calculate the Jaccard index again:

$$\begin{aligned} |P| = 2018 \quad |P \cap T| = 597 \qquad |P \cup T| = 3670 \qquad J(P,T) = 0.1627 \\ |P \cap U| = 243 \qquad |P \cup U| = 3254 \qquad J(P,U) = 0.0747 \\ |P \cap T \cap U| = 116 \quad |P \cup T \cup U| = 4758 \quad J(P,T,U) = 0.0244 \end{aligned}$$

The precision of the P-method is now $p'_P = 0.2096$, thus much closer to $p_T = 0.2885$. Recall does not change for obvious reasons. The new F-score is $f'_P = 0.2666$. We conclude that the T-method and the P-method work almost equally well but are still outperformed, at least for SWT, by a simple list of words not being in a general language dictionary.

[5] Engl. *NN is called ADJ if*

5.4 Comparison with *Sketch Engine*

We also extract terms from the corpus with the keyword extraction method provided by *Sketch Engine* (Kilgarriff et al., 2014), for further comparison. These terms have not been considered for the candidate list given to the raters because this extraction was done after the raters' work. Thus, the results are not completely comparable to the others.

We extract 1000 MWT and 1000 SWT with *Sketch Engine* with a minimum frequency of 1 to create conditions comparable to those of the P-method. The reference corpus for the term extraction by *Sketch Engine* is the German Web 2013 (deTenTen13) (Jakubíček et al., 2013).

Sketch Engine finds 198 terms which were not in the list given to the raters. 139 of them are SWT and 59 MWT. One of the raters annotates these 198 items with the criteria which resulted from the discussion. 65.94 % of the SWT and 35.59 % of the MWT are considered as useful for the dictionary. However, some of the selected SWT already appear in our candidate list as a part of MWT because the different tools use different criteria to distinguish between SWT and MWT.

We also calculate the Jaccard index between the results of the three different tools introduced in Section 4 and the list provided by *Sketch Engine*. S stands for the *Sketch Engine* in the calculations given below. We use the original P-list without the above-mentioned modifications.

$$
\begin{array}{lll}
|P \cap S| = 177 & |P \cup S| = 4895 & J(P,S) = 0.0362 \\
|T \cap S| = 108 & |T \cup S| = 4140 & J(T,S) = 0.0261 \\
|U \cap S| = 76 & |U \cup S| = 3402 & J(U,S) = 0.0223 \\
|P \cap T \cap U \cap S| = 0 & |P \cup T \cup U \cup S| = 7535 & J(P,T,U,S) = 0
\end{array}
$$

Now, we only take those terms into consideration which were selected for the final lemma list:

$$
\begin{array}{lll}
|P_s \cap S_s| = 62 & |P_s \cup S_s| = 545 & J(P_s,S_s) = 0.1138 \\
|T_s \cap S_s| = 22 & |T_s \cup S_s| = 724 & J(T_s,S_s) = 0.0304 \\
|U_s \cap S_s| = 59 & |U_s \cup S_s| = 873 & J(U_s,S_s) = 0.0676 \\
|P_s \cap T_s \cap U_s \cap S_s| = 0 & |P_s \cup T_s \cup U_s \cup S_s| = 1453 & J(P_s,T_s,U_s,S_s) = 0
\end{array}
$$

The results show that the terms extracted by *Sketch Engine* are closest to those extracted by the P-method, but the Jaccard index is still under 0.1 and only slightly above 0.1 for the selected terms. All the values here are below those calculated above.

6. Conclusion and future work

The described methods led to the definition of the final lemma list for creating the electronic dictionary on graph theory. Which information is given in the microstructure of a particular lemma is defined by its category. For example, the entry of a lemma from the category PERSON provides information on THEOREMS named after this PERSON, whereas a lemma from the category TYPES gives the information which PROPERTIES this TYPE of graphs has or can have. The information needed to provide such items will also be extracted by means of patterns and interactive corpus exploration.

The objective of our study was to compare the output of different term extractors, to understand to which degree such output can be used as a lemma list of the dictionary, and which amount of post-processing is needed to end up with an adequate lemma list. We note that a combination of different techniques may still be needed to cover the domain adequately. And two lessons are, if not learned from the exercise, at least recapitulated: deciding on termhood is also hard for experts, as long as no very strict guidelines are given; and lexicographic lemma selection also depends on the lexicographer's intuition about the dictionary's target group as well as on their strategy to ensure a homogeneous treatment of lexical items with respect to lemma selection.

With a view to further automating the lemma selection process, one could suggest a comparison of the term extraction with existing lemma lists for the domain; but such list do not really exist for graph theory terminology in German. One approach could be to use the titles of articles in Wikipedia which belong to the category *Graphentheorie*[6], but this list only comprises 100 items and thus is on a totally different scale than the amounts in our work. Furthermore, such a comparison does not take the available corpus into account; its results would thus only be significant to a very limited extent.

Not only the methods to identify lemma and item candidates, but also the evaluation methods are adaptable to other mathematical fields. This way it becomes easier to create electronic LSP-dictionaries for mathematical domains.

For selecting the lemmas, we worked with a German corpus. As we also have a comparable corpus of English texts, we will experiment with a similar (semi-)automatic approach for English.

To answer the question how the methods can be improved to also extract the terms which were only given in the U-list requires further research. In the end, we can see that a combination of different term extraction tools might work best because their pairwise Jaccard index is really low. We will take these results into consideration when working with the English data. Nevertheless, an extra difficulty is that we are only interested in the terminology of a sub-domain, not of a whole domain. Thus, some issues remain although we have chosen our corpus data according to this prerequisite.

7. References

Amjadian, E., Inkpen, D., Paribakht, T. & Faez, F. (2016). Local-Global Vectors to Improve Unigram Terminology Extraction. In *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 2–11. URL https://www.aclweb.org/anthology/W16-4702.

Amjadian, E., Inkpen, D., Paribakht, T.S. & Faez, F. (2018). Distributed specificity for automatic terminology extraction. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 24(1), pp. 23–40. URL https://www.jbe-platform.com/content/journals/10.1075/term.00012.amj.

Bernier-Colborne, G. (2012). Defining a gold standard for the evaluation of term extractors. In *Proceedings of the Terminology and Knowledge Representation Workshop, LREC 2012,*. pp. 15–18.

[6] https://de.wikipedia.org/wiki/Kategorie:Graphentheorie

Blum, A. & Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98. New York, NY, USA: Association for Computing Machinery, pp. 92–100. URL https://doi.org/10.1145/279943.279962.

Bohnet, B. (2010). Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pp. 89–97. URL https://www.aclweb.org/anthology/C10-1011.

Bonin, F., Dell'Orletta, F., Montemagni, S. & Venturi, G. (2010). A Contrastive Approach to Multi-word Extraction from Domain-specific Corpora. In N.C.C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner & D. Tapias (eds.) *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), pp. 3222–3229.

Cabre, M.T. & Vivaldi Palatresi, J. (2013). 110. Acquisition of terminological data from text: Approaches. In R.H. Gouws, U. Heid, W. Schweickard & H.E. Wiegand (eds.) *Supplementary Volume Dictionaries. An International Encyclopedia of Lexicography*. De Gruyter Mouton, pp. 1486–1497. URL https://doi.org/10.1515/9783110238136.1486.

Choi, J.D., Tetreault, J. & Stent, A. (2015). It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 387–396. URL https://www.aclweb.org/anthology/P15-1038.

Conrado, M., Pardo, T. & Rezende, S. (2013). A Machine Learning Approach to Automatic Term Extraction using a Rich Feature Set. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*. Atlanta, Georgia: Association for Computational Linguistics, pp. 16–23. URL https://www.aclweb.org/anthology/N13-2003.

Dobrov, B. & Loukachevitch, N. (2011). Multiple Evidence for Term Extraction in Broad Domains. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*. Hissar, Bulgaria: Association for Computational Linguistics, pp. 710–715. URL https://www.aclweb.org/anthology/R11-1103.

Fedorenko, D., Astrakhantsev, N. & Turdakov, D. (2014). Automatic Recognition of Domain-Specific Terms: an Experimental Evaluation. *Proceedings of the Institute for System Programming of RAS*, 26(4), pp. 55–72. URL https://doi.org/10.15514/ispras-2014-26(4)-5.

Fleiss, J.L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), pp. 378–382. URL http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1972-05083-001&lang=de&site=ehost-live.

Frantzi, K., Ananiadou, S. & Mima, H. (2000). Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2), pp. 115–130. URL https://doi.org/10.1007/s007999900023.

Frantzi, K.T. & Ananiadou, S. (1996). Extracting Nested Collocations. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. pp. 41–46. URL https://www.aclweb.org/anthology/C96-1009.

Hätty, A. (2020). *Automatic term extraction for conventional and extended term definitions across domains*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart. URL http://elib.uni-stuttgart.de/handle/11682/11136.

Heid, U. & Weller, M. (2010). Corpus-derived data on German multiword expressions for lexicography. In A. Dykstra & T. Schoonheim (eds.) *Proceedings of the 14th EURALEX International Congress*. Leeuwarden/Ljouwert, The Netherlands: Fryske Akademy, pp. 331–340.

Jaccard, P. (1902). Lois de distribution florale dans la zone alpine. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 28(144), pp. 69–130.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P. & Suchomel, V. (2014). Finding Terms in Corpora for Many Languages with the Sketch Engine. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 53–56. URL https://www.aclweb.org/anthology/E14-2014.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P. & Suchomel, V. (2013). The TenTen Corpus Family. In *7th International Corpus Linguistics Conference CL 2013*. Lancaster, pp. 125–127. URL http://ucrel.lancs.ac.uk/cl2013/.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P. & Suchomel, V. (2014). The Sketch Engine: ten years on. *Lexicography*, pp. 7–36.

Kochetkova, N.A. (2015). A method for extracting technical terms using the modified weirdness measure. *Automatic Documentation and Mathematical Linguistics*, 49(3), pp. 89–95. URL https://doi.org/10.3103/s0005105515030036.

Kruse, T. & Giacomini, L. (2019). Planning a domain-specific electronic dictionary for the mathematical field of graph theory: definitional patterns and term variation. In I. Kosem, T.Z. Kuhn, M. Correia, J.P. Ferreira, M. Jansen, I. Pereira, J. Kallas, M. Jakubíček, S. Krek & C. Tiberius (eds.) *Electronic lexicography in the 21st century. Proceedings of the eLex 2019 conference*. Brno: Lexical Computing CZ, s.r.o., pp. 676–693.

Kruse, T. & Heid, U. (2020). Lemma Selection and Microstructure: Definitions and Semantic Relations of a Domain-Specific e-Dictionary of the Mathematical Domain of Graph Theory. In Z. Gavriilidou, M. Mitsiaki & A. Fliatouras (eds.) *Euralex Proceedings*, volume 1. pp. 227–233.

Landis, J.R. & Koch, G.G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), pp. 159–174. URL http://www.jstor.org/stable/2529310.

Meyer, I. (2001). Extracting knowledge-rich contexts for terminography: A conceptual and methodological framework. In D. Bourigault, C. Jacquemin & M.C. L'Homme (eds.) *Recent Advances in Computational Terminology*, volume 2 of *Natural Language Processing*. Amsterdam/Philadelphia: John Benjamins, pp. 279–302.

Pennington, J., Socher, R. & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL https://www.aclweb.org/anthology/D14-1162.

Pollak, S., Repar, A., Martinc, M. & Podpečan, V. (2019). Karst Exploration: Extracting Terms and Definitions from Karst Domain Corpus. In I. Kosem, T.Z. Kuhn, M. Correia, J.P. Ferreira, M. Jansen, J.K. Isabel Pereira, M. Jakubíček, S. Krek & C. Tiberius (eds.) *Electronic lexicography in the 21st century. Proceedings of the eLex 2019 conference*. Sintra: Lexical Computing CZ s.r.o, pp. 934–956.

Rayson, P. & Garside, R. (2000). Comparing Corpora using Frequency Profiling. In *The Workshop on Comparing Corpora*. Hong Kong, China: Association for Computational Linguistics, pp. 1–6. URL https://www.aclweb.org/anthology/W00-0901.

Rigouts Terryn, A., Hoste, V., Drouin, P. & Lefever, E. (2020). TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. In *Proceedings of the 6th International Workshop on Computational Terminology*. Marseille, France: European Language Resources Association, pp. 85–94. URL https://www.aclweb.org/anthology/2020.computerm-1.12.

Roelcke, T. (2010). *Fachsprachen*. Berlin: Erich Schmidt, 3rd, newly revised edition edition.

Roesiger, I., Bettinger, J., Schäfer, J., Dorna, M. & Heid, U. (2016). Acquisition of semantic relations between terms: how far can we get with standard NLP tools? In *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 41–51. URL https://www.aclweb.org/anthology/W16-4706.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester.

Schmid, H. & Laws, F. (2008). Estimation of Conditional Probabilities With Decision Trees and an Application to Fine-Grained POS Tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, pp. 777–784.

Schäfer, J., Rösiger, I., Heid, U. & Dorna, M. (2015). Evaluating noise reduction strategies for terminology extraction. In *Proceedings of the conference Terminology and Artificial Intelligence 2015 (Granada, Spain)*. Granada, Spain: Universidad de Granada, pp. 123–131.

Wang, R., Liu, W. & McDonald, C. (2016). Featureless Domain-Specific Term Extraction with Minimal Labelled Data. In *Proceedings of the Australasian Language Technology Association Workshop 2016*. Melbourne, Australia, pp. 103–112. URL https://www.aclweb.org/anthology/U16-1011.

Zadeh, B.Q. & Handschuh, S. (2014). The ACL RD-TEC: A Dataset for Benchmarking Terminology Extraction and Classification in Computational Linguistics. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 52–63. URL https://www.aclweb.org/anthology/W14-4807.

This work is licensed under the Creative Commons Attribution ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-sa/4.0/