Terminological Saturation in Retrospective Text Document Collections

Cross-Evaluation of Automated Term Extraction Tools

Authors:

Victoria Kosa (ZNU),

David Chaves-Fraga (UPM),

Dmitriy Naumenko (BWT),

Eugene Yuschenko (BWT),

Carlos Badenes-Olmedo (UPM)

Vadim Ermolayev (ZNU),

Aliaksandr Birukou (SPRINGER)

30.09.2017

Document Information

Project Information:					
Project ID:		**Acronym:**	TS-RTDC		
Full Title:	Terminological Saturation in Retrospective Text Document Collections				
Project URL:	N/A				
Project Manager:	Vadim Ermolayev	**Partner:**	ZNU	**E-mail:**	vadim@ermolayev.com

Deliverable Information:					
Type:	Technical Report			Planned	Actual
Status:	Version 0.4 Draft		**Date of Delivery:**	30.09.2017	22.09.2017
Dissemination: (to check)	Public: ☑	Consortium: □	**Document ID:**	TS-RTDC-TR-2017-1	
URL:	http://ermolayev.com/TS-RTDS-TR-2017-1.pdf				

Author Information:			
Authors (Partners):	Victoria Kosa (ZNU), David Chaves Fraga (UPM), Dmitriy Naumenko (BWT), Eugene Yuschenko (BWT), Carlos Badenes (UPM), Vadim Ermolayev (ZNU), and Aliaksandr Birukou (SPRINGER)		
Responsible Author:	Victoria Kosa	**Partner:**	ZNU
		E-mail:	v.kosa@znu.edu.ua

Document Information:	
Abstract: (for dissemination)	This document reports on our activity in cross-evaluating the two freely available software tools for automated term extraction (ATE) from English texts: NaCTeM TerMine and UPM Term Extractor. The objective to do this cross evaluation was to find the most fitting software for extracting the bags of terms to be the part of our instrumental pipeline for exploring terminological saturation in professional text document collections in a domain of interest. The choice of these particular tools from the bunch of the other available is explained in our review of the related work in ATE. The approach to measure terminological saturation is based on the use of the THD algorithm developed in frame of our OntoElect methodology for ontology refinement. The report presents the suite of instrumental software modules, experimental workflow, 2 synthetic and 3 real document collections, generated datasets, and the set-up of our experiments. The results of the cross-evaluation experiments are further presented, analyzed, and discussed. Finally the report offers some conclusions and recommendations on the use of ATE software for measuring terminological saturation in retrospective text document collections.
Keywords:	Automated Term Extraction, Software Tool, Experimental Cross-Evaluation, Terminological Saturation
Citation:	Kosa, V., Chaves-Fraga, D., Naumenko, D., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., and Birukou, A.: Cross-Evaluation of Automated Term Extraction Tools. Technical Report TS-RTDC-TR-2017-1, 30.09.2017, Dept of Computer Science, Zaporizhzhia National University, Ukraine, 60 p.

Revision Log:

Version/Revision	**Date**	**Change(s)**	**Submitted by**	**Comment**
0.2, Draft	31.07.2017	**Initial** draft containing experimental set-up and partial results	Victoria Kosa	
0.3, Draft	15.09.2017	**Intermediate** draft: SOTA-methods, all experimental results, updated references	Victoria Kosa	
0.4, Draft	17.09.2017	**Intermediate** draft. Added: - Sect. Executive Summary - Sect 8: Conclusions and Recommendations A few language and style corrections done.	Vadim Ermolayev	
0.5, Draft	20.09.2017	**Pre-final** Draft. Added: - Sect 2.2 Available Software Implementations (draft) Corrected: Experimental W-flow in Fig. 4 Layouts and formats checked and corrected	Vadim Ermolayev, Victoria Kosa	
0.6, Final	22.09.2017	**Final**. Added: - Links to instrumental software modules	Dmitriy Naumenko, David Chaves-Fraga, Vadim Ermolayev	

Authors and Affiliations

Victoria Kosa and **Vadim Ermolayev**

Department of Computer Science
Zaporizhzhya National University (**ZNU**)
Zhukovskogo st. 66,
69600, Zaporizhzhia
Ukraine

David Chaves-Fraga and
Carlos Badenes-Olmedo

Ontology Engineering Group,
Universidad Politécnica de Madrid (**UPM**)
Madrid,
Spain

Dmitriy Naumenko and **Eugene Yuschenko**

BWT Group (**BWT**)
Mayakovskogo st. 11,
69035, Zaporizhzhia,
Ukraine

Aliaksandr Birukou

Springer-Verlag GmbH (**SPRINGER**)
Tiergartenstrasse 17,
69121, Heidelberg,
Germany

Table of Contents

Executive Summary

This document reports on our activity in cross-evaluating the two freely available software tools for automated term extraction from English texts: NaCTeM TerMine and UPM Term Extractor. The objective to do this cross evaluation was to find the most fitting software for extracting the bags of terms to be the part of our instrumental pipeline for exploring terminological saturation in professional text document collections in a domain of interest. Hence, we designed and set up these cross-evaluation experiments as if the tools are used in this pipeline. The choice of these particular tools from the bunch of the other available is explained in our review of the related work in Section 2.

The approach to measure terminological saturation is based on the use of the THD algorithm developed in frame of our OntoElect methodology for ontology refinement. This part of OntoElect is outlined in Section 3.

Sections 4 to 7 of the report present our contributions.

Section 4 formulates our research questions towards evaluating the influence of different factors on terminological saturation. Further, a generic workflow is developed to support the four different series of experiments on studying the influence of these factors. Our experimental set-up is then explained, based on the generic workflow.

Section 5 presents the suite of instrumental software modules which has been developed to support our experimental workflow. These software modules cover all the steps which are too laborious to be performed manually: the extraction of document collections from the Web; document conversion and generation of the datasets; extraction of the terms; and computing terminological difference to assess saturation.

Section 6 presents the document collections and datasets, and further elaborates on the details of the experimental set-up. For evaluating the aspect of the choice of term extraction software, we have prepared and used two synthetic and three real document collections of full-text papers from different domains.

The results of our cross-evaluation experiments are presented and discussed in Section 7.

Finally, the summarization of the results in the form of conclusions and recommendations is given in Section 8, which concludes the report. The findings suggest that the use of **UPM Extractor is preferred** over TerMine to detect **terminological saturation** or **excessive noise**. This use is not constrained by a subject domain and does not depend on manual de-noising of the source data in the collection.

List of Figures

List of Tables

1 Introduction

Automated term extraction (ATE, also known as recognition – ATR) from textual documents is an established sub-field in text mining. Its results are further used for different important purposes, for example as inputs in ontology learning. Many research activities are undertaken currently to improve the quality of extraction results. These activities focus on different aspects, including: new or improved extraction algorithms; combining linguistic and statistical approaches to extraction; developing new or refined metrics which allow higher quality extraction; developing new extraction tools which yield better results and scale to fit current dataset size requirements. The mainstream criteria used to assess the quality of extracted results are adopted from information retrieval and based on recall and precision metrics. However, to the best of our knowledge, there were no reports on approaches to assess the completeness of the document collection from which extraction is performed. Recall measures just inform about how completely the set of terms was extracted from the available data but does not hint if the data itself was complete to contain all important terms characterizing the domain. In other words, there is no way so far to check if the collection of documents chosen for term extraction is representative. Therefore the approaches to measure the representativeness of document collections are timely. In this context, it is also important to know what would be a minimal representative subset of documents.

The research presented in this report[1] develops the methodological and instrumental components for measuring the representativeness of high-quality collections of textual documents. It is assumed that the documents in a collection cover a single and well circumscribed domain and have a timestamp associated with them – so can be ordered by publication time. A typical example of such a collection is the set of the full text papers of a professional journal or conference proceedings series. The main hypothesis, put forward in this work, is that a sub-collection can be considered as representative to describe the domain, in terms of its terminological footprint, if any additions of extra documents from the entire collection to this sub-collection do not noticeably change this footprint. Such a sub-collection is further considered as complete and could be used e.g. for learning an ontology from it. In fact, this approach to assess the representativeness does so by evaluating terminological saturation in a document collection.

Of course, in this approach we are concerned about automated term extraction, as doing so manually is not feasible for any realistic document collection pretending to cover a professional domain. Therefore, it is important to know if terminological saturation depends on a term extraction method, implemented in a software tool. For finding this out, the presented research project cross-evaluated the two software tools. The choice of these particular tools from the bunch of the other

[1] This research is performed as the PhD project by the first author. Its exposé has been presented in [1].

available is explained in our review of the related work in Section 2.

The approach to measure terminological saturation is based on the use of the THD algorithm developed in frame of our OntoElect methodology for ontology refinement [2]. This part of OntoElect is outlined in Section 3.

Sections 4 to 7 present our contributions.

We formulated our research questions towards evaluating the influence of different factors on terminological saturation. Further, we developed a generic workflow to support the four different series of experiments on studying the influence of these factors. We provided a more detailed experimental set-up, based on the generic workflow, for studying the influence of the choice of the term extraction software. This contribution is presented in Section 4.

We developed the suite of instrumental software modules to support our experimental workflow. The instrumental software covers all the steps which are too laborious to be done manually: the extraction of document collections from the Web; document conversion and generation of the datasets; extraction of the terms; and computing terminological difference to assess saturation. This contribution is presented in Section 5.

For evaluating the aspect of the choice of a term extraction software, we cross-evaluated the two selected software tools, UPM Term Extractor[2] versus NaCTeM TerMine[3], on two synthetic and three real document collections of full-text papers from different domains. Section 6 presents the document collections and datasets, and further elaborates on the details of the experimental set-up. The results of our cross-evaluation experiments are presented and discussed in Section 7.

Finally, we summarize our results, in the form of conclusions and recommendations in Section 8, which concludes the report.

[2] UPM Term Extractor could be downloaded from https://github.com/ontologylearning-oeg/epnoi-legacy. It has to be further installed locally for use..

[3] The batch service of NaCTeM TerMine is available at http://www.nactem.ac.uk/batch.php. Access needs to be requested.

2 Motivation and Related Work

Extracting terminology from texts is a complicated and laborious process which requires a substantial part of highly qualified human effort. Despite that, it is more and more often used in many important applications, e.g. for engineering ontologies [2], [3]. So, knowing the smallest possible representative document collection for a domain is very important to efficiently develop ontologies with satisfactory domain coverage. Therefore, laying out a method to determine a terminologically saturated subset of documents of the minimal size within a collection is topical. It is also important to make this method as efficient and automated as possible to lower the overhead on the core knowledge engineering workflow.

As the focus of this report is to find out which relevant term extraction software yields the best (smallest) saturated sub-sets of documents, we review the related work along the following aspects. We look at the comparison of existing ATE approaches in terms of the quality of their results. We also consider as relevant those methods (ATE algorithms plus metrics) which are domain-independent, unsupervised, and allow assessing the significance of extracted terms. Further we check if the selected methods are implemented as software tools which are publicly available for our experiments. We also pay attention to whether the tools return data for term significance evaluations that are essential for our further saturation measurements.

2.1 Methods for Automated Term Extraction

Despite being important for practice, ATE is still far from being reliable. New approaches to ATE are being proposed and still demonstrate their precision at the level below 80 percent [4]. So, these can hardly be used in industry. Several reviews have been performed to compare and cross-evaluate ATE methods, e.g. [5]. Perhaps, [4] and [20] are the most recent work on that.

In the majority of approaches to ATE, e.g. [6] or [7], processing is done in two consecutive phases: Linguistic Processing and Statistical Processing. Linguistic processors, like POS taggers or phrase chunkers, filter out stop words and restrict candidate terms to n-gram sequences: nouns or noun phrases, adjective-noun and noun-preposition-noun combinations. Statistical processing is then applied to measure the ranks of the candidate terms. These measures are [5] either the measures of ‘unithood’, which focus on the collocation strength of units that comprise a single term; or the measures of ‘termhood’ which point to the association strength of a term to domain concepts.

For ‘unithood’, the metrics are used such as: mutual information [8], log likelihood [9], t-test [6], [7], and the notion of ‘modifiability’ and its variants [10], [7]. The metrics for ‘termhood’ are either term frequency-based (unsupervised approaches) or reference corpora-based (semi-supervised approaches). The most used frequency–based metrics are: TF/IDF (e.g. in [4], [11]); weirdness [12] which compares the frequency of a term in the evaluated corpus with that in the reference corpus; domain pertinence [14]. More recently, hybrid approaches were proposed, that combine 'unithood' and 'termhood' measurements in a single value. A representative metric is c/nc-value [13]. C/nc-value-based approaches to ATE have received their further evolution in many works, e.g. [6], [14], [15] – to mention a few.

Linguistic Processing is organized and implemented in a very similar fashion in all the ATE methods, except for some of them also include filtering out stop words. Stop words (terms) could be filtered out also at a cut-off step after statistical processing. So, in our review and selection we further look at the second phase of Statistical Processing only. Statistical Processing is sometimes further split in two consecutive sub-phases: term candidate scoring, and ranking. For term candidates scoring, reflecting its likelihood of being a term, known methods could be distinguished by being based on (c.f. [4]): measuring occurrences frequencies (including word association); assessing occurrences contexts; using reference corpora, e.g. Wikipedia [16]; topic modeling [17].

The cut-off procedure, takes the top candidates, based on scores, and thus distinguishes significant terms from insignificant (or non-) terms. Many cut-off methods rely upon the scores, coming from one scoring algorithm, and establish a threshold in one or another way. Some others that collect the scores from several scoring algorithms use (weighted) linear combinations [18], voting [5], [2], or (semi-)supervised learning [19]. In our set-up, we do cut-offs after term extraction based on voting, as explained in Section 3. So, the ATE algorithms / solutions which perform cut-offs together with scoring are not relevant for our experimental setting.

Based on the evaluations in [5], [4], [20] the most widely used ATE algorithms, for which their performance assessments are published, are listed in Table 1. The table also provides the assessments on the aspects we use for selection.

Table 1: The comparison of the most widely used ATE metrics and algorithms

Method [Source]	**Domain-independence (+/-)**	**Super-vizion (U/SS)**	**Metrics**	**Term Signi-ficance**	**Cut-off (+/-)**	**Precision (GENIA; average)**	**Run Time (%/c-value)**
TTF [21]	+	U	Term (Total) Frequency	+	-	0.70; 0.35	0.34
ATF [20]	+	U	Average Term Frequency	+	-	0.71; 0.33	0.37
						0.75; 0.32	0.35
TTF-IDF [22]	+	U	TTF+Inverse Document Frequency	+	-		
						0.82; 0.51	0.35
RIDF [23]	+	U	Residual IDF	-		0.71; 0.32	0.53
						0.80; 0.49	0.37
C-value [13]	**+**	**U**	c-value,	**+**	**-**	0.73; **0.53**	1.00

Method [Source]	**Domain-independence (+/-)**	**Super-vizion (U/SS)**	**Metrics**	**Term Significance**	**Cut-off (+/-)**	**Precision (GENIA; average)**	**Run Time (%/c-value)**
			nc-value			0.77; **0.56**	1.00
Weirdness [12]	+/-	SS	Weirdness	-		0.77; 0.47	0.41
						0.82; 0.48	1.67
GlossEx [18]	+	SS	Lexical (Term) Cohesion, Domain Specificity	-			
						0.70; 0.41	0.42
TermEx [14]	+	SS	Domain Pertinence, Domain Consensus, Lexical Cohesion, Structural Relevance	-	+		
						0.87; 0.46	0.52
PU-ATR [16]	-	SS	nc-value, Domain Specificity	-	+	0.78; 0.57	809.21

Comments:

Domain Independence: "+" stands for a domain-independent method; "-" marks that the method is either claimed to be domain-specific by its authors, or is evaluated only on one particular domain. We are looking for a domain-independent method.

Supervision: "U" – unsupervised; "SS" – semi-supervised. We are looking for an unsupervised method.

Term Significance: "+" – the method returns a value for each retained term which could further be used as a measure of its significance compared to the other terms. "-" – marks that such a measure is not returned or the method does the cut-off itself. We are looking for receiving a measure to do cut-offs later.

Cut-off: "+" – the method does cut-offs itself and returns only significant terms; "-" – the method does not do cut-offs. We are looking for "-".

Precision and Run Time: The values are based on the comparison of the two cross-evaluation experiments reported in [4] / [20]. Empty cells in the table mean that there was no data for this particular method in this particular experiment. [4] used ATR4S – open-source software written in Scala. It evaluated 13 different methods, implemented in ATR4S, on 5 different datasets, including GENIA. [20] used JATE 2.0, free software written in Java. It evaluated 9 different methods, implemented in JATE, on 2 different datasets, including GENIA. So, the results on GENIA are the baseline for comparing the Precision. Two values are given for each reference experiment: precision on GENIA; average precision. Both [4] and [20] experimented with c-value method which was the slowest on average for [20]. So, the execution times for c-value were used as a baseline to normalize the rest in the Performance column.

After looking at Table 1, we support the conclusion of [20] *c-value* is the most reliable method as it obtains consistently good results, in terms of precision, both on the two different mixes of datasets – [20] and [4]. We also note that *c-value* is one of the slowest methods among the group of unsupervised and domain-independent, though its performance is comparable with the fastest ones. Still, *c-value* outperforms the domain-specific methods, sometimes significantly – as it is in the case with PU-ATR.

Hence, we have chosen *c-value* as the method for our cross-evaluation experiments. We will therefore be looking at the tools which implement *c-value* and are publicly freely available.

2.2 Available Software Implementations

For choosing the software tools that implement the *c-value* method for ATE we looked at the implementations of term extraction tools at several popular web resources like at http://inmyownterms.com/terminology-extraction-tools/ or https://en.wikipedia.org/wiki/Terminology_extraction. In addition to the reference implementations mentioned before: ATR4S [4] and JATE 2.0 [20], we have identified the following freely available ATE software tools as indicated in Table 2.

Table 2: Free ATE Software Tools (Listed Alphabetically)

Name / Owner	Website	Short description	Algorithm / Metric	Domain	Constraints
BioTex / LIRMM	http://tubo.lirmm.fr/biotex/	extracts biomedical terms from free text		Bio-medical	Domain-specific
FiveFilters / Medialab-Prado	http://fivefilters.org/term-extraction/	extracts terms through a web service; relies on a PHP port of Topia's Term Extraction; a simple alternative to Yahoo Term Extraction service	Occurrence (TTF) and word count in a term	independent	Web service, size of text constrained
TaaS (Terminology as a Service EU Project)	https://term.tilde.com/	Identify term candidates in your documents and extract them automatically. Uses CollTerm (linguistic) or Kilgray (statistical) services	Frequency-based	independent	Does not provide term significance scores
TerMine / NaCTeM	http://www.nactem.ac.uk/software/termine/	Extracts terms from plain English texts, provides the Batch mode (access to be requested for non-UK academic users)	*c-value*	independent	The service requests to avoid heavy bulk processing
TermFinder / Translated.net	https://labs.translated.net/terminology-extraction/	A Web application that extracts terms from the inserted text. Compares the frequency of words in a given document with their frequency in the language (generic corpus).	Poisson statistics, Maximum Likelihood Estimation and IDF	requires language corpus	Returns the score of a term as a numeric value (%)
TBXTools [24] / Universitat Oberta de Catalunya	https://sourceforge.net/projects/tbxtools/	A Python toolset using NLTK (Natural Language Toolkit)	TTF	Independent, multilingual, requires language corpus	Deletes n-grams woth stop words

Name / Owner	Website	Short description	Algorithm / Metric	Domain	Constraints
UPM Term Extractor [25] / Dr Inventor EU project	https://github.com/ontologylearning-oeg/epnoi-legacy	A Java software for extracting terms and relations from scientific papers	*c-value*	Independent	Takes text input data of at most 15 Mb

For the final selection of the tools for our cross-evaluation we:

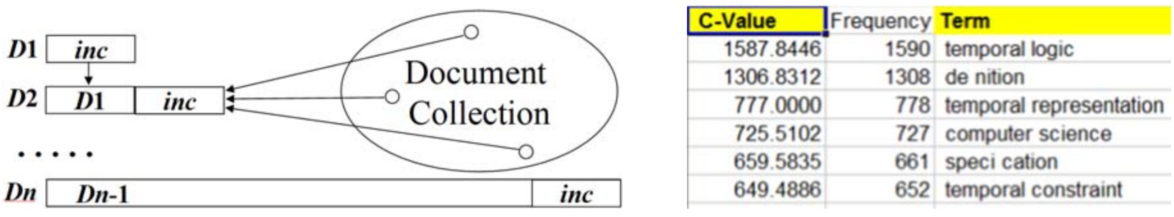
- Decided not to consider ATR4S and JATE 2.0, at list at this stage, because it was not fully clear how to extract the *c-value* method implementation from these suites
- Selected the tools that use the *c-value* method – which are NaCTeM TerMine and UPM Term Extractor

3 OntoElect Saturation Metric and Measurement Pipeline

OntoElect, as a methodology, seeks for maximizing the fitness of the developed ontology regarding what the domain knowledge stakeholders think about the domain. Fitness is measured as the stakeholders' "votes" – a metric that allows assessing the stakeholders' commitment to the ontology under development – reflecting how well their sentiment about the requirements is met. The more votes are collected – the higher the commitment is expected to be. If a critical mass of votes is acquired (say 50%+1, which is a simple majority vote), the ontology is considered to satisfactorily meet the requirements.

It is well known that direct acquisition of requirements from domain experts is not very realistic as they are expensive and not really willing to do the work falling out of their core activity. So, in this project, we are focused on the indirect collection of the stakeholders' votes by extracting these from high quality and reasonably high impact documents authored by the stakeholders.

An important feature to be ensured for knowledge extraction from text collections is that the dataset needs to be statistically representative to cover the opinions of the domain knowledge stakeholders satisfactorily fully. OntoElect suggests a method to measure the terminological completeness of the document collection by analyzing the *saturation* of terminological footprints of the incremental slices of the document collection – as e.g. reported in [26]. The full texts of the documents from the retrospective collection are grouped in datasets in the order of their timestamps. As pictured in Fig. 1a, the first dataset *D*1 contains the first portion (*inc*) of documents. The second dataset *D*2 contains the first dataset *D*1 plus the second incremental slice (*inc*) of documents. Finally, the last dataset *Dn* contains all the documents from the collection.



C-Value	Frequency	Term
1587.8446	1590	temporal logic
1306.8312	1308	definition
777.0000	778	temporal representation
725.5102	727	computer science
659.5835	661	specification
649.4886	652	temporal constraint

(a) (b)

Fig. 1: (a) Incrementally enlarged datasets in OntoElect; (b) an example of a bag of terms extracted by TerMine.

At the next step of the OntoElect workflow the bags of multi-word terms *B*1, *B*2, …, *Bn* are extracted from the datasets *D*1, *D*2, …, *Dn*, using TerMine software, together with their *significance* (C-value) scores. Those scores correlate to a significant extent to term frequencies – i.e. how often a term was met in the dataset. Please see an example of a bag of terms extracted by TerMine in Fig. 1b.

At the subsequent step, every extracted bag of terms *Bi*, *i* = 1, …, *n* is processed as follows:

- **Normalized scores** are computed for each individual term: *n-score* = *C-value* / max(*C-value*)
- **Individual term significance threshold** (*eps*) is computed to cut off those terms that are not within the majority vote. The sum of *n-scores* having values above *eps* form the majority vote if this sum is higher that ½ of the sum of all *n-scores*.
- The **cut-off** at *n-score* < *eps* is done
- The result is saved in *Ti*

After this step only significant terms, whose *n-scores* represent the majority vote, are retained in the bags of terms. *Ti* are then evaluated for saturation by measuring pair-wise terminological difference between the subsequent bags *Ti* and *Ti*+1, *i* = 0, …, *n*-1. It is done by applying the THD algorithm [2]. We provide it also here in Fig. 2 for completeness.

```
Algorithm THD. Compute Terminological Difference between Bags of Terms
   Input: $T_i, T_{i+1}$                                   Pick up one
   Output: $thd(T_i, T_{i+1})$
   $sum := 0$
   for $k = 1, \|T_{i+1}\|$                                 Look for linguistically similar in the previous
      $sum := sum + ns_k^{i+1}$
      $ident := .F.$                                       Found: check the n-scores
      for $m = 1, \|T_i\|$
            if $(t_m^i, t_k^{i+1})_\equiv$ then do $thd := thd + |ns_m^i - ns_k^{i+1}|$; $ident := .T.$ end do
      end for
      if $ident := .F.$ then $thd := thd + |ns_k^{i+1}|$
   end for                                                 Not found: add the n-score
   $thdr := thd / sum$
```

Fig. 2: THD algorithm [2] for comparing a pair of bags of terms. It has been modified, compared to [2], for computing the *thdr* value.

In fact, THD accumulates, in the *thd* value for the bag *Ti*+1, the *n-score* differences if there were linguistically the same terms in *Ti* and *Ti*+1. If there was no the same term in *Ti*, it adds the *n-score* of the orphan to the *thd* value of *Ti*+1. After *thd* has been computed, the relative terminological difference *thdr* receives its value as *thd* divided by the sum of *n-scores* in *Ti*+1.

Absolute (*thd*) and relative (*thdr*) terminological differences are computed for further assessing if *Ti*+1 differs from *Ti* more than the individual term significance threshold *eps*. If not, it implies that adding an increment of documents to *Di* for producing *Di*+1 did not contribute any noticeable amount of new terminology. So, the subset *Di*+1 of the overall document collection may have become terminologically saturated. However, to obtain more confidence about the saturation, OntoElect suggests that some more subsequent pairs of *Ti* and *Ti*+1 are evaluated. If stable saturation is observed, then the process of looking for a minimal saturated sub-collection could be stopped. Sometimes, however, a terminological peak may occur after saturation has been observed in the previous pairs of *T*. Normally this peak indicates that a highly innovative document with a substantial number of new terms has been added in the increment. An example of saturation evaluation for the TIME document collection [26] using OntoElect is pictured in Fig. 3.

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
empty-D1	8546	3439	3.0000	838	54.4448	100.0000
D1-D2	14597	5654	3.1699	1179	35.9807	62.3806
D2-D3	23992	9223	3.7549	1548	36.0855	59.6366
D3-D4	31427	12127	4.0000	2104	23.7044	35.4153
D4-D5	38122	14923	4.7549	2183	22.4341	30.7901
D5-D6	42788	17256	5.0000	2400	14.9911	18.7218
D6-D7	49986	20127	5.0000	2821	17.4853	20.7287
D7-D8	59294	23950	5.0000	3430	23.1877	26.9035
D8-D9	65627	26418	5.0000	3767	13.1819	15.3747
D9-D10	75171	30473	5.6147	3584	25.0810	36.7663
D10-D11	81617	33296	6.0000	3893	9.6005	13.8278
D11-D12	91692	37637	6.0000	4410	13.3894	19.7595
D12-S13	101190	41803	6.0000	4903	9.0502	12.6376
D13-D14	108203	44655	6.0000	5255	7.3260	9.8946
D14-D15	115493	47792	6.0000	5658	8.5976	11.7790
D15-D16	121832	50475	6.0000	6007	6.6174	9.0302
D16-D17	128171	53180	6.3043	5564	6.3422	9.0829
D17-D18	137918	57368	6.3399	6043	13.0734	20.2061
D18-D19	145173	60346	6.3549	6109	5.1033	8.0395
D19-D20	151075	62839	6.6667	6259	5.4895	8.7677

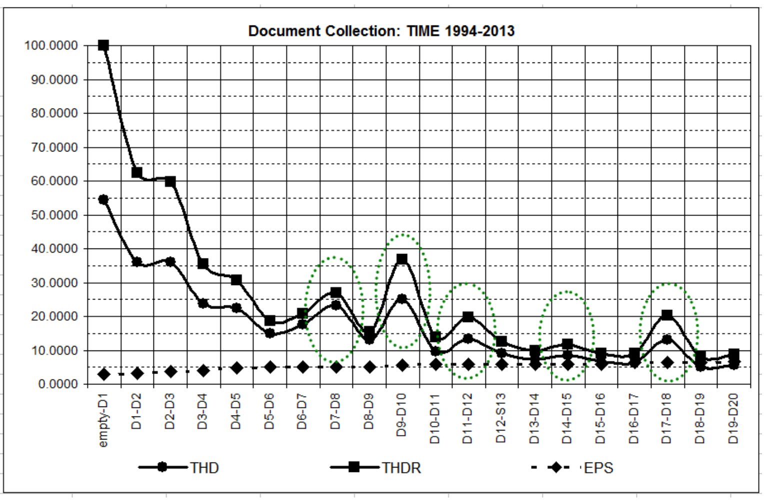


Fig. 3: The results of evaluating the saturation of the TIME Collection (adapted from [26]). Terminological peaks are observed at D7-D8, D9-D10, D14-D15, D17-D18. As explained in [26], the peaks are correlated with the added frequently cited papers. It is also worth noticing that the number of retained terms in *T*-s is significantly lower that the number of extracted terms in *B*-s.

To finalize this brief presentation of the OntoElect approach for assessing terminological saturation, it is worth noting that it is domain independent and unsupervised – due to the use of TerMine for term extraction. However, the dependence of OntoElect on this tool implies a substantial shortcoming – is able to process only English documents. More shortcomings are revealed in our experimental study discussion (Section 7). One of the tasks for our research, on which we focus in this paper, is trying OntoElect pipeline with the alternative term extraction tool – UPM Term Extractor – and cross-evaluate the results versus those obtained using NaCTeM TerMine.

4 Research Questions and Experimental Workflow

The objective of the presented experimental research project is to check if the OntoElect approach to assess the representativeness of a subset within a document collection, based on measuring terminological saturation, is valid. The setting of the experiments should consider several aspects which may influence the measurements and, therefore the results of measuring saturation.

4.1 Research Questions

These aspects are taken into account while answering the following research questions:

Q1: Which of the term extraction software tools yield better saturated sets of terms?

Q2: Which would be the proper direction in forming the datasets to check saturation: chronological, reverse-chronological, bi-directional, random selection? Which direction is the most appropriate to cope with potential terminological drift in time?

Q3: Would the size of a dataset increment influence saturation measurements? Is there an optimal size of an increment for the purpose?

Q4: Would frequently cited documents form a minimal representative subset of documents? Do the most frequently cited documents indeed provide the biggest terminological contribution to the document collection?

Q5: Is the method for assessing completeness based on saturation measurements valid? Does it indeed provide a correct indication of statistical representativeness?

The answers to the outlined research questions **Q1** – **Q4** are sought based on conducting experiments using different document collections coming from different domains and communities.

Perhaps, **Q5** is the most difficult question to answer and it still requires some thinking for offering a convincing method to assess the adequacy and validity of the experimental method investigated in the presented project. One possible way is to do that based on the cross-evaluation with another method for ontology learning, e.g. [27]. Another possible way is to select a much smaller subset of a document collection, e.g. only the papers with high terminological impact. The set of terms extracted from this “decisive minority vote” subset could be manually checked by human experts.

4.1 Generic Experimental Workflow

The experimental workflow, outlined in Fig. 4, is based on the OntoElect processing pipeline described in Section 3. This workflow could be generically applied (using Configure Experiment step) to perform all the experimental series described below.

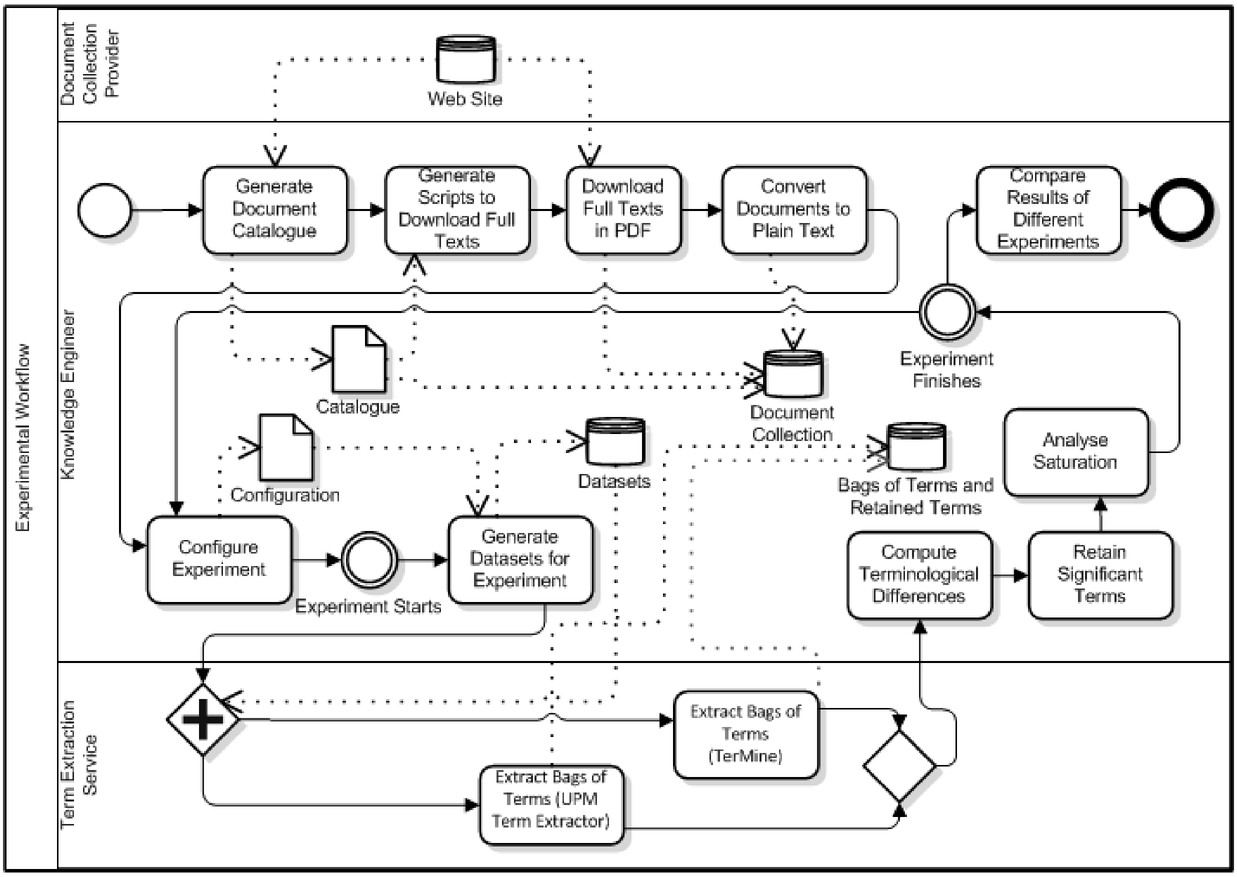

Experimental Workflow
Document Collection Provider
Knowledge Engineer
Term Extraction Service
Web Site
Generate Document Catalogue
Generate Scripts to Download Full Texts
Download Full Texts in PDF
Convert Documents to Plain Text
Compare Results of Different Experiments
Catalogue
Document Collection
Experiment Finishes
Configuration
Datasets
Bags of Terms and Retained Terms
Analyse Saturation
Configure Experiment
Experiment Starts
Generate Datasets for Experiment
Compute Terminological Differences
Retain Significant Terms
Extract Bags of Terms (TerMine)
Extract Bags of Terms (UPM Term Extractor)

Fig. 4: Experimental workflow

The workflow covers the preparatory phase, experiment configuration, the generation of the datasets, term extraction, saturation measurement, and the analysis of the results. Some of the steps in these phases can only be performed manually, like Configure Experiment, Analyze Saturation, and Compare Results. These steps are not too laborious, however, and the effort does not noticeably grow with the number of documents. The rest of the steps require instrumental software support, especially for large document collections.

The **preparatory** phase includes:

- Generation of the catalogue for the chosen document collection using the information available at the publisher's web site. This catalogue includes all the metadata for the documents, including their abstracts, and also the numbers of their citations acquired from Google Scholar4.
- Download of the full texts of the papers, usually in PDF format, based on the information in the catalogue. This step may require the permission granted by the owner of the collection to bulk-download their full texts.
- Conversion of the full texts of the downloaded documents to the plain text format for further term extraction.

[4] http://scholar.google.com/

The **configuration** phase is the choice of the experimental setting and the parameters of the datasets to be generated. The experimental setting is defined by the series – i.e. by the research question we wish to answer. The parameters are hence defined by the objective of the series. These parameters are: the order of adding documents to a dataset, the size of an increment, the software tool used for term extraction.

The **datasets generation** phase takes these parameters and the document collection in the plain text format. The datasets are then generated, which will further be taken by term extraction. The datasets are built as described in Section 3. The texts are added to the increments in the order taken as the parameter of the experiment.

The phase of **term extraction** consequently applies the chosen software tool to the generated datasets: *D*1, *D*2, … In result, it outputs the bags of extracted terms *B*1, *B*2, …

The **saturation measurement** phase applies the THD algorithm to the bags of terms as explained in Section 3. It outputs the results in the tabular form as, for example pictured in Fig. 3.

The **analysis** and **comparison** are done manually using any appropriate software tool. We use MS Excel in our experiments.

Our experimental workflow is fully covered by the developed and used software as described in Section 5.

4.2 Planned Series of Experiments

Different series of experiments, using this workflow, are planned to be conducted in the presented study.

The **first series** are planned for experimental cross-evaluation of the available alternative term extraction software tools. Based on the datasets with the increments of reasonable size, term extraction is done separately using the UPM Term Extractor and NaCTeM TerMine. The results are compared in terms of saturation measures. This may allow answering **Q1**.

The **second series** of experiments will be targeted at checking which order of choosing papers for the datasets yields better saturated sets of terms and assesses terminological temporal drift. In this series the experimental workflow will use the term extraction software selected in series 1 and be applied to the datasets which are formed: (i) chronologically; (ii) reverse-chronologically; (iii) bi-directionally, i.e. including data increments containing the documents from both ends of the temporal span in turns (e.g. first issue, than last issue, than second issue, etc.); and (iv) including documents picked from the data collection uniformly randomly. Saturation measures and saturated sets of terms will be compared across these different choices. This series will allow answering **Q2**.

To answer **Q3**, the **third series** will focus on finding out what might be the optimal size of an increment to form experimental datasets. For this series, the datasets will be formed following the optimal paper selection direction discovered in the second series. The size of the increments will however be varying. Saturation measurements will be compared for different data increment sizes and the optimal value will be discovered if such an optimum does exist.

The **fourth series** will base on the most appropriate paper selection order, determined in the second series, and investigate the terminological impact of the frequently cited documents in the collection. For that, the impact of each document will be computed based on its citation frequency. The documents with impact equal to *n* will be replicated *n* times in the corresponding dataset. The experimental workflow will be applied to these "impact" datasets and the results will be compared to the first series using "flat" datasets. The comparison will be done in terms of saturation measures and terminological contribution peaks [26]. This experiment may allow to answer **Q4** and extract the "decisive minority vote" subset of terms contributed by the high-impact papers, as e.g. been done in [26] for Time Representation and Reasoning domain.

4.3 Cross-Evaluation of Selected ATE Software Tools

In this report we focus on answering **Q1**. For this we conducted the first series of experiments to cross-evaluate UPM Term Extractor versus NaCTeM TerMine. In this sub-section we present the configuration of these experimental series and the measurements in more detail.

We did this cross-evaluation by applying the experimental workflow to the two synthetic and three selected real document collections coming from different domains. Before applying the tools to the real document collections we check if they perform adequately on the two specifically crafted synthetic collections representing the boundary cases – for immediate saturation and no saturation. The first synthetic collection contains just one paper and the datasets grow by adding replicas of this paper as increments. In this case, both tools should return saturated bags of terms very quickly as all the increments are terminologically the same. The second synthetic collection is for checking the opposite case – the documents are all different and come from different fields. In this case the tools are expected to deliver results which do not saturate terminologically. All the document collections are presented in more detail in Section 6.

To cross-evaluate term extraction tools we look at:

- How quickly the bags of terms, extracted from the incrementally growing datasets, saturate terminologically in terms of *thd* versus *eps*. We also measure *thdr*. The results are measured for all the document collections, independently for each tool, and then compared.
- If the tools extract statistically similar bags of terms from each of the document collections. The similarity between the extracted bags of terms is also measured using *thd* versus *eps* approach by applying the THD algorithm (and software tool) to the pairs (*B*1, *B*1*m*), (*B*2, *B*2*m*), …, (*Bn*, *Bnm*), where

Bi is the bag of terms extracted by the first chosen tool (UPM Term Extractor) and *Bim* is the bag of terms extracted by the second chosen tool (NaCTeM TerMine). The intuition behind this measurement is that, if the tools extract similar sets of terms with similar *c-values*, then the terminological difference (*thd*) between such bags of terms will be low.

5 Instrumental Software

The **preparatory** phase of our experimental workflow is supported by the following three software modules.

Catalogue Generator. We found out in the pre-implementation phase that developing a generic module for creating the catalogue of the papers is not feasible due to the layout differences at different publisher resources. For example, the journal and proceedings pages, from which the information about the papers needs to be parsed, look very differently for Springer, IEEE, or ACM. Therefore we opted to develop tailored parsers. In the reported project, a tailored parser[5] for Springer journal pages has been developed. The parser takes a Springer Link journal web page URL as its input and stores the list of all the papers of this journal in the specified .csv file[6]. The information about a paper contains all its reference information, the abstract, and the no of citations acquired from Google Scholar.

Full Text Downloader. For downloading the full texts of the papers another software module has been developed[7]. It receives a .csv list of papers to be downloaded and generates a script to download the full texts of the papers based on their DOI information taken from the catalogue. The papers in PDF are stored in a folder specified as a parameter. The PDF files are named, using the information from the catalogue, as follows: <journal_ID>+”-”+<year>+<vol>+”(”+<issue>+”)-(”+<pages>+”)-”+<DOI>+”.pdf”

PDF to Plain Text Convertor. One more software module has been developed[8] for batch conversions of paper full texts in PDF to plain text. It gets a path to the directory where PDF documents are stored, as a parameter. It produces the outputs for each input file in plain text (ANSI) format in which hyphenations are removed and each sentence occupies a separate line for better term extraction.

All these modules are command line PHP tools.

The **datasets generation** phase of our experimental workflow is supported by the **Dataset Generator** module[9]. This module takes the following inputs:

[5] The catalogue extractor for Springer journals is available at: https://github.com/bwtgroup/SSRTDC-Springer-article-parser.

[6] The catalogues of the acquired journal papers for the KM collection in .XLSX format are available at: https://github.com/bwtgroup/SSRTDC-PaperCatalogues/. The data has been collected on December 3-4, 2016.

[7] The PDF downloader is available at: https://github.com/bwtgroup/SSRTDC-Collections-Springer-PDF-Downloader

[8] The PDF to plain text convertor is available at: https://github.com/bwtgroup/SSRTDC-PDF2TXT

[9] The dataset generator is available at: https://github.com/bwtgroup/SSRTDC-PDF2TXT

1. **PFOLDER** – the name of the folder containing the TXT documents for forming the datasets. It assumes that the text files in this folder are named using the following convension: <journal_ID>+"-"+<year>+<vol>+"("+<issue>+")-("+<pages>+")-"+<DOI>+".txt". Hence, the information about the time of publication (timestamp) is encoded in the name of a file: <year>+<issue>.
2. **ORDER** – the order in which the documents are picked to be added to datasets. Four different values are possible: (i) "chrono" for the chronological order; (ii) "rev-chrono" for counter-chronological order; (iii) "bi-dir" for bi-directional order; and (iv) "random" for picking the documents randomly.
3. **INCRSIZE** – the number of papers to be included in a dataset increment
4. **DFOLDER** – the name of the folder to store the generated datasets

The datasets are formed following the OntoElect procedure described in Section 3. One topical difference to OntoElect is that the papers could be added following the four different orders.

The **term extraction** phase is supported by the two software tools: **UPM Term Extractor** and **NaCTeM TerMine**.

UPM Term Extractor has been developed in the Dr Inventor project. The tool takes an English (PDF or plain text) corpus of documents and returns the bag of extracted terms as a CSV file. Each term is provided in a separate line with its *c-value*.

NaCTeM TerMine is a publicly available service which is used in a batch mode[10]. It takes an English plain text (ANSI) document as a file to upload and returns the bag of extracted terms as a CSV output. Each term is provided in a separate line and accompanied with its numeric *c-value* and *frequency*. The tool allows choosing a text parser from the list of the two available options: tree tagger and genia tagger.

The **saturation measurement** and **analysis** phases are supported by the **THD** modules, the **Convertor** module, and **Stop Term Remover** module.

The **THD modules** have been developed in Python to implement the THD[11] algorithm for the input bags of terms in UPM Term Extractor and NaCTeM TerMine formats. The modules process the sequence of the pairs of the bags of terms as presented in Section 3. The bags of terms have to be stored in separate plain text files. The list of the files to be processed is taken from the "list.txt" configuration file.

[10] Batch mode for TerMine is freely accessible at http://www.nactem.ac.uk/batch.php for academic purposes, provided that the permission by NaCTeM is granted for non-UK users.

[11] The THD modules are available at: https://github.com/bwtgroup/SSRTDC-modules/tree/master/THD

In addition to these two THD implementations, we also developed the **Convertor**[12] which takes a bag of terms in TerMine format and saves it in the UPM Extractor format. This module was needed as a pre-processor for the THD module for the case when the outputs from different extraction tools are checked for being identical.

Finally, the **Stop Term Remover[13]** module has been developed to lower the effort needed to remove the set of manually selected stop terms from the datasets. It takes the list of the manually selected terms in a plain text input file and deletes all these terms from the bags of terms extracted either by Termine or UPM Extractor.

Only two modules in the suite are constrained by some specifics in data. The **Catalogue Generator** is tailored to **Springer Link journal pages** – so it is Document Collection dependent. **NaCTeM TerMine** service and **UPM Term Extractor** take only **English texts**. The rest of the software modules can be used to process any Document Collection, coming from an arbitrary domain, and in any language.

[12] The bags of terms convertor is available at: https://github.com/bwtgroup/SSRTDC-modules/tree/master/BTC

[13] The stop term remover is available at: https://github.com/bwtgroup/SSRTDC-modules/tree/master/STR

6 Document Collections and Datasets

In this section we describe the data used in our experiments. These data come from two synthetic and three real document collections. The synthetic collections are 1DOC and RAW. The real document collections are TIME, DMKD, and DAC.

6.1 Synthetic Document Collections

Our synthetic collections have been prepared to evaluate the boundary cases: one in which terminological saturation should happen immediately; and the other one in which terminological saturation should not happen. These cases help us evaluate if saturation metric is adequate in these extreme cases. If so, there is more confidence that it is adequate for real document collections.

1DOC is the document collection containing just one paper. We used the source of [24]. It has been converted to plain ANSI text format manually. From the plain text, the datasets $D1$, $D2$, …, $D20$[14] have been generated, as described in Section 3, and the increment for each subsequent dataset was the text of this one paper. So, $D1$ contained one copy of this paper text, $D2$ – two copies of the same text, …, $D20$ – 20 copies of the same text. It is straightforward that, if the OntoElect approach to measuring saturation is correct, the saturation in this case should be observed starting already at comparing $T1$ and $T2$ with thd very close to 0. The reason for that is that all the increments are identical.

The intuition behind assembling the RAW collection is opposite to the previous case. To avoid saturation, we need a collection in which all the increments are substantially terminologically different – so that the subsets of significant terms in the incremental slices always have $thd > eps$. To have that, we need to put together the documents dealing with different topics, coming from different fields, and therefore using very different terminology. For getting such a collection of documents we have randomly selected 80 articles from English Wikipedia such that no two of them are about a similar topic and the size of an article is not too small. The articles have been downloaded in 1-column PDF format. Further, we processed these PDF files to convert into plain ASCII texts using our PDF to Plain Text Convertor (Section 5). The texts have not been cleaned to keep the possibility for checking how does the noise injected by Wikipedia into the PDF printouts influences saturation. Based on the plain texts, we generated 20 datasets, $D1$, $D2$, …, $D20$[15], with increments comprising 4 randomly selected documents from this collection.

[14] The **1DOC** collection in plain text and the datasets generated of these texts are available at: https://www.dropbox.com/sh/64pbodb2dmpndcy/AABFp9lZKwJE9A5X_9VbSHa-a/1DOC?dl=0

[15] The **RAW** collection in plain text and the datasets generated of these texts are available at: https://www.dropbox.com/sh/64pbodb2dmpndcy/AAAyDghrEkmlX0dpNh-zzDB3a/RAW?dl=0

6.2 Real Document Collections

Our real document collections are all composed of the high quality papers published at the peer-reviewed international venues in three different domains:

- The TIME collection contains the full text papers of the proceedings of the TIME Symposia series[16]
- The DMKD collection contains the subset of full text articles from the Springer journal on Data Mining and Knowledge Discovery[17]
- The DAC collection contains the subset of full text papers of the Design Automation Conference[18]

The domain of the TIME collection is Time Representation and Reasoning. The publisher of these papers is IEEE. This collection has been acquired in our previous research reported in [24]. The complete TIME collection contains all the papers published in the TIME symposia proceedings between 1994 and 2013, which are 440 full text documents in total. The papers of the TIME collection have been processed manually, including their conversion to plain texts and cleaning of these texts. So, the resulting datasets were not very noisy. We have chosen the increment for generating the datasets to be 20 papers. So, based on the available texts, we have generated 22 incrementally enlarged datasets $D1$, $D2$, …, $D22$[19].

The domain of DMKD collection is Data Mining and Knowledge Discovery, which falls into our broader target domain of Knowledge Management as its essential part. This collection is also the part of a broader KM collection presented in [1] and provided by Springer based on their policy on full text provision for data mining purposes[20]. For the KM collection, based on their expert advice, fifteen Springer journals[21] have been selected that are broadly relevant to the domain of Knowledge Management. Knowledge Management has been chosen as a target domain because: (i) the methodology developed in the presented experimental study is for knowledge engineering and management; (ii) the partners in the presented project possess extensive expertise in Knowledge Management and therefore could be used as subject experts; (iii) there is a substantially big collection of high-quality full text documents broadly relevant to this domain

[16] http://time.di.unimi.it/TIME_Home.html

[17] https://link.springer.com/journal/10618

[18] http://dac.com/

[19] The **TIME** collection in plain text and the datasets generated of these texts are available at: https://www.dropbox.com/sh/64pbodb2dmpndcy/AAAzVW7aEpgW-JrXHaCEqg2Sa/TIME?dl=0

[20] https://www.springer.com/gp/rights-permissions/springer-s-text-and-data-mining-policy/29056

[21] The list of the selected journals is available at: https://github.com/bwtgroup/SSRTDC-PaperCatalogues/blob/master/ListOfJournals.xls

available at Springer. Overall, the KM collection appears to be well suited to attack all the research questions outlined in Section 4. Indeed, it is formed of the journals scoping into different subfields of Computer Science, related to Knowledge Management. The journals in the selection are also mutually complementary in terms of providing terminology related to Knowledge Management. So, there seems to be a balance between the broadness of the overall scope and the focus on the target domain. This balance needs to be checked experimentally by verifying if it contains a saturated terminological footprint on the domain. Furthermore, individual journal collections chronologically start at very different times and contain quite different numbers of volumes, issues and papers. So, these internal dis-balances may help reveal the complications like terminological temporal drift and different terminological contributions caused by varying data volumes coming from different journals. The composition of the KM collection is diagrammatically shown in Fig. 5.

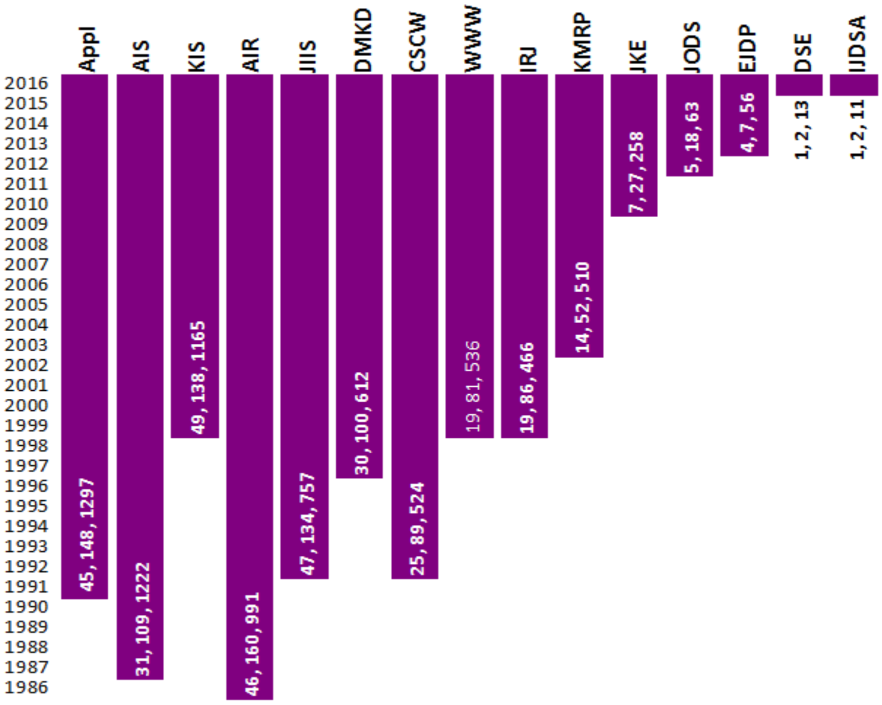


Fig. 5: Distribution of papers in the journals of the KM document collection. Y-axis shows the years of publication, X-axis corresponds to the journals. The numbers in the bars are: no of volumes, no of issues, the total no of papers in the journal.

For our cross-evaluation (research question **Q1**), we have taken the subset of KM because the full collection is unnecessary big for the task. To the DMKD document collection, we have included 300 papers belonging to just one of the fifteen journals – Data Mining and Knowledge Discovery. These papers have been published between 1997 and 2010. All the papers in their full texts were automatically processed using our instrumental pipeline presented in Section 5. In difference to the TIME collection, no manual cleaning of document texts was applied. For generating the datasets, the increment has been chosen to be 15[22] papers. So, based on the available documents we have generated 30 incrementally enlarged datasets *D*1, *D*2, …, *D*30[23].

The domain of the DAC collection is Engineering Design Automation. The publisher of these papers is IEEE. For the collection, we have chosen 506 papers published between 2004 and 2010. The papers of the DAC collection have been automatically converted to plain text using our instrumental software. We deliberately skipped manual cleaning of the plain texts to be able to compare the results between very noisy (DAC) and not very noisy (TIME) datasets generated from the papers having the same publisher and, therefore, the same source layout (IEEE). Similarly to TIME, we have chosen the increment for generating the datasets to be 20 papers. So, based on the available texts, we have generated 26 incrementally enlarged datasets *D*1, *D*2, …, *D*26[24].

6.3 Summary of Data Features

The characteristics of all the five document collections and datasets are summarized in Table 3 below.

Table 3: The features of the used document collections and datasets

Collec-tion	Type	Paper Type and Layout	No Doc	Noise	Processing	Inc	No Datasets
1DOC	synthetic	journal, ACM 1-column	1	manually cleaned	manual	1 paper	20
RAW	synthetic	Wikipedia 1-column	80	not cleaned, moderately noisy	automated	4 papers	20
TIME	real	conference, IEEE 2-column	437	manually cleaned	manual conversion to plain text, automated dataset generation	20 papers	22
DMKD	real	journal, Springer 1-column	300	not cleaned, moderately noisy	automated	15 papers	15
DAC	real	conference, IEEE 2-column	506	not cleaned, quite noisy	automated	20 papers	26

For all real collections, the documents have been added to the datasets in their chronological order of publication. For the RAW collection the documents have been added in random order.

[22] Which yields roughly similar to TIME increment sizes, as the journal papers in DMKD are bigger than TIME conference papers.

[23] The **DMKD** collection in plain text and the datasets generated of these texts are available at: https://www.dropbox.com/sh/64pbodb2dmpndcy/AAAsLqmyWVPGTFe_7KRpXkeJa/DMKD?dl=0

[24] The **DAC** collection in plain text and the datasets generated of these texts are available at: https://www.dropbox.com/sh/64pbodb2dmpndcy/AABb7SaxWDzPaWdsYF_7MpSca/DAC?dl=0

7 Experiments and Discussion

In this section we report the results of our experiments on the datasets generated from all the five data collections, as presented in Section 6, particularly on the results of the phases of term extraction, saturation measurement, and analysis and comparison. We also discuss these results. The experiments have been set and performed using the workflow and instrumental tools presented in Sections 4 and 5.

In the experiment with each collection we:

- Extracted the bags of terms from the prepared datasets using: (a) NaCTeM TerMine; and (b) UPM Term Extractor
- Measured saturation for both sets of the bags of terms using the corresponding THD modules
- Measured comparative saturation for the pairs of the bags of terms ($B1$, $B1m$), ($B2$, $B2m$), …, (Bn, Bnm) – as described in Section 4.3
- Built the diagrams and analyzed the results

In addition to the above activities, for the RAW collection we also looked at the effect of removing stop terms after doing term extraction. By removing these stop terms, which represented the injection of noise by Wikipedia and also the text fragments from the figures, we de-noised the output. The lists of the stop words (terms) were prepared manually based on the extractions from the last dataset $D20$. These stop terms were further automatically removed from all the datasets using our Stop Term Remover module. So, for the RAW collection we also compared noisy and cleaned bags of terms.

We first report the results of measuring saturation for our synthetic document collections – 1DOC and RAW. We then analyze the results for our real collections – DMKD, TIME, and DAC.

7.1 Terminological Saturation in Synthetic Collections

Per collection design, as described in Section 6, the results on the 1DOC collection are expected to demonstrate immediate saturation and the results on the RAW collection have to be quite far from being saturated.

For the bags of terms extracted from the 1DOC collection the results of measuring saturation look as follows.

We first processed the bags of terms extracted by TerMine. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) are presented in Table 4. These measurements are visualized in the diagram of Fig. 6(a).

Table 4: Saturation measurements of the 1DOC bags of terms extracted by NaCTeM TerMine

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	2159	570	2.4	149	21.9460797	100
D1-D2	2159	1204	4	347	11.5402114	34.4624956
D2-D3	2159	1207	**6**	**347**	**2.57E-06**	**7.69E-06**
D3-D4	2159	1208	8	347	2.36E-06	7.04E-06
D4-D5	2159	1208	10	347	1.37E-06	4.08E-06
D5-D6	2159	1208	12	347	1.15E-06	3.43E-06
D6-D7	2159	1208	14	347	1.39E-06	4.16E-06
D7-D8	2159	1208	16	347	1.24E-06	3.71E-06
D8-D9	2159	1208	18	347	2.49E-06	7.45E-06
D9-D10	2159	1208	20	347	1.01E-06	3.01E-06
D10-D11	2159	1208	22	347	1.03E-06	3.06E-06
D11-D12	2159	1208	24	347	1.63E-06	4.88E-06
D12-D13	2159	1208	26	347	8.01E-07	2.39E-06
D13-D14	2159	1208	28	347	9.25E-07	2.76E-06
D14-D15	2159	1208	30	347	1.39E-06	4.15E-06
D15-D16	2159	1208	32	347	2.45E-06	7.32E-06
D16-D17	2159	1208	34	347	1.39E-06	4.16E-06
D17-D18	2159	1208	36	347	1.14E-06	3.40E-06
D18-D19	2159	1208	38	347	1.92E-06	5.74E-06
D19-D20	2159	1208	40	347	1.11E-06	3.33E-06

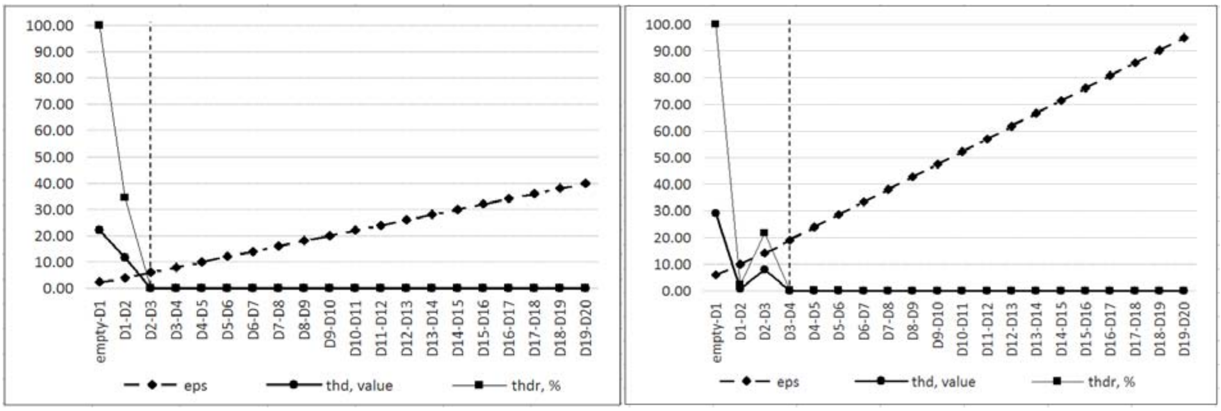
The dashed vertical line in Fig. 6(a) points to the bag of terms (extracted from *D*3) in which saturation indicator has been observed for the first time as *thd* went below *eps*. In fact, and as expected, we further observe very stable saturation with the same number of extracted terms and increasing individual term significance thershold *eps*. The values of *thd* and *thdr* drop down to become statistically equal to zero starting from *T*2-*T*3.

We then measured the terminological differences between the bags of terms extracted by UPM Extractor. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) are presented in Table 5. These measurements are visualized in the diagrams of Fig. 6(b).

Table 5: Saturation measurements for the 1DOC bags of terms extracted by UPM Term Extractor

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	4422	2200	6.000000	393	29.007090	100.000000
D1-D2	4422	2451	10.000000	412	0.679515	2.306007
D2-D3	4422	3019	14.264663	707	8.072802	21.564555
D3-D4	4422	3019	**19.019550**	**707**	**0.074083**	**0.198122**

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
D4-D5	4422	3019	23.774438	707	0.067454	0.180291
D5-D6	4422	3019	28.529325	707	0.052147	0.139487
D6-D7	4422	3019	33.284213	707	0.035010	0.093619
D7-D8	4422	3019	38.039100	707	0.037906	0.101426
D8-D9	4422	3019	42.793988	707	0.034140	0.091310
D9-D10	4422	3019	47.548875	707	0.028485	0.076206
D10-D11	4422	3019	52.303763	707	0.021185	0.056665
D11-D12	4422	3019	57.058650	707	0.028822	0.077135
D12-D13	4422	3019	61.813538	707	0.023589	0.063101
D13-D14	4422	3019	66.568425	707	0.018214	0.048731
D14-D15	4422	3019	71.323313	707	0.018736	0.050131
D15-D16	4422	3019	76.078200	707	0.018865	0.050483
D16-D17	4422	3019	80.833088	707	0.016440	0.043981
D17-D18	4422	3019	85.587975	707	0.017158	0.045916
D18-D19	4422	3019	90.342863	707	0.012301	0.032911
D19-D20	4422	3019	95.097750	707	0.016213	0.043390



(a) Bags of terms extracted by TerMine (b) Bags of terms extracted by UPM Extractor

Fig. 6: Visualization of saturation measurements on the 1DOC datasets

The dashed vertical line in Fig. 6(b) points to the bag of terms (extracted from *D*4) in which saturation indicator has been observed for the first time as *thd* went below *eps*. Very similarly to the case of TerMine, and as expected, we further observed very stable saturation with the same number of extracted terms and increasing individual term significance threshold *eps*. The values of *thd* and *thdr* drop down to become statistically equal to zero starting from *T*3-*T*4.

The differences in saturation measurements for the bags of terms extracted by TerMine and UPM Extractor are as follows:

- UPM Extractor generated bigger bags of terms with *c-value* > 1: 3 019 terms versus 1 208 in the TerMine case
- Individual term significance thresholds (*eps*) were about 2.5 times higher for UPM Extractor

- The number of retained terms with *c-value* > *eps* was approximately 2 times bigger in the UPM Extractor case
- The values of *thd* and *thdr* were significantly lower (~10 000 times) for TerMine

Overall, TerMine results showed a quicker convergence to saturation than that by UPM Extractor. From the other hand: (i) the number of retained terms from the saturated sub-collection; and (ii) the cut-off point at the individual term significance threshold were higher in the UPM Extractor results. Based on observing these differences, we can conclude that, linguistically, TerMine was circa 3 times more selective in extracting term candidates. So, the pre-processing in TerMine is more sophisticated and, probably, more accurate. From the other hand, the cut-offs in UPM Extractor outputs happened for approximately two times more significant terms. Hence, the statistical processing part in UPM Extractor circumscribes more compact, yet significant sets of terms. This points out that, due to the statistical processing phase, UPM Extractor is a more precise instrument.

We further checked if both tools extracted statistically similar sets of terms from the 1DOC collection. The measurements are presented in Table 6 and visualized in Fig. 7.

Table 6: Comparison of the retained sets of terms extracted from 1DOC collection by UPM Term Extractor and NaCTeM TerMine

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
D1-D1m	5547	2515	4.754888	587	8.972861	28.668056
D2-D2m	5547	3100	**9.509775**	569	**2.291409**	7.717779
D3-D3m	5547	3668	14.264663	575	2.430730	8.149254
D4-D4m	5547	3668	19.019550	569	2.224458	7.504159
D5-D5m	5547	3668	23.774438	569	2.111576	7.121521
D6-D6m	5547	3668	28.529325	569	2.100859	7.089746
D7-D7m	5547	3668	33.284213	575	2.284566	7.665941
D8-D8m	5547	3668	38.039100	575	2.278613	7.649792
D9-D9m	5547	3668	42.793988	569	2.086197	7.040108
D10-D10m	5547	3668	47.548875	569	2.083446	7.032331
D11-D11m	5547	3668	52.303763	569	2.081196	7.023752
D12-D12m	5547	3668	57.058650	575	2.267164	7.613153
D13-D13m	5547	3668	61.813538	575	2.267091	7.609816
D14-D14m	5547	3668	66.568425	575	2.264937	7.603543
D15-D15m	5547	3668	71.323313	569	2.073309	6.999081
D16-D16m	5547	3668	76.078200	575	2.262151	7.595876
D17-D17m	5547	3668	80.833088	569	2.073279	6.998189
D18-D18m	5547	3668	85.587975	569	2.069935	6.988349
D19-D19m	5547	3668	90.342863	575	2.260311	7.588357
D20-D20m	5547	3668	95.097750	569	2.069381	6.987496

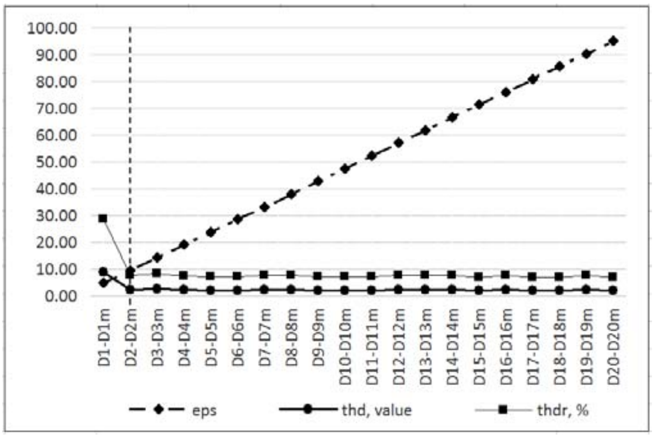


Fig. 7: Comparison of the retained sets of terms extracted from the 1DOC collection by UPM Term Extractor and NaCTeM TerMine

Table 6 and Fig. 7 show that both tools extracted statistically identical bags of terms despite the fact that the numbers of retained terms differed significantly in the individual cases (reported above). The terminological difference became statistically negligible at the second measurement point, where the *thd* value (2.291409) went significantly below *eps* (9.509775). This situation was stable, since the *thd* values oscillated around 2.1 and the *eps* values steadily went up to 95.

For the bags of terms extracted from the RAW collection the results of measuring saturation look as follows.

We first processed the bags of terms extracted by TerMine. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) are presented in Table 7. These measurements are visualized in the diagram of Fig. 8(a).

Table 7: Saturation measurements of the RAW bags of terms extracted by NaCTeM TerMine

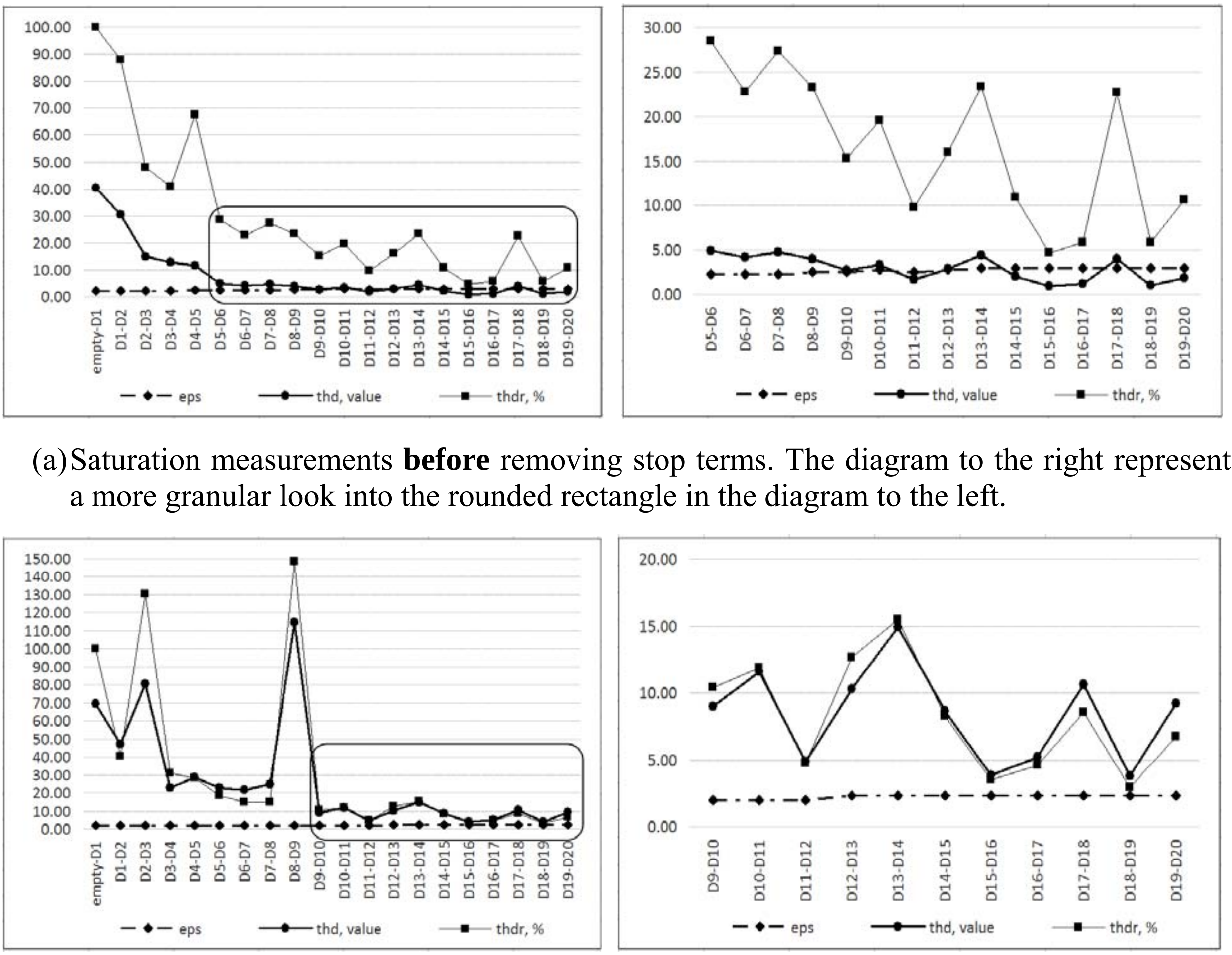
Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	7530	1455	2.000000	788	40.449828	100.000000
D1-D2	11984	2432	2.000000	1301	30.444777	87.814134
D2-D3	14696	3018	2.000000	1602	15.056950	47.971707
D3-D4	19406	3939	2.000000	2080	12.744844	40.818641
D4-D5	29165	5327	2.321928	1321	11.596120	67.574677
D5-D6	35029	6389	2.321928	1607	4.916363	28.492079
D6-D7	39601	7271	2.321928	1866	4.133303	22.812089
D7-D8	44015	8267	2.321928	2126	4.796644	27.361703
D8-D9	49954	9608	2.584963	2057	3.956798	23.293184
D9-D10	56024	10543	2.584963	2315	2.681278	15.284548
D10-D11	60718	11656	2.800000	2360	3.335927	19.554684
D11-D12	63477	12314	2.584963	2710	1.729368	9.765197
D12-D13	70824	13871	2.807355	2828	2.902100	15.992738

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
D13-D14	78688	15956	3.000000	3144	4.413035	23.328680
D14-D15	84017	17118	3.000000	3376	2.081457	10.860068
D15-D16	87156	17828	3.000000	3527	0.912837	4.715075
D16-D17	97961	18626	3.000000	3696	1.148129	5.826125
D17-D18	104892	20250	3.000000	4031	4.034517	22.649926
D18-D19	107830	20810	3.000000	4152	1.026730	5.839607
D19-D20	116209	22015	3.000000	4449	1.877501	10.630587

We then analyzed *B*20, extracted by TerMine, going from the top of the list down to the terms having *c-value*s greater than 40. Based on this scan, we extracted the list of circa 200 stop terms. These stop terms have been removed from the bags of terms *B*1, …, *B*20 and saturation analysis has been repeated then. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) for so de-noised bags of terms are presented in Table 8. These measurements are visualized in the diagrams of Fig. 8(b).

Table 8: Saturation measurements of the RAW bags of terms, extracted by NaCTeM TerMine, after removing stop terms

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	7470	1418	2.000000	754	69.471692	100.000000
D1-D2	11907	2384	2.000000	1255	47.205952	40.458438
D2-D3	14615	2963	2.000000	1548	80.600214	130.511939
D3-D4	19323	3879	2.000000	2022	22.854872	31.204546
D4-D5	28836	5244	2.000000	2770	28.718382	28.166183
D5-D6	34699	6303	2.000000	3359	22.782218	18.263364
D6-D7	39087	7180	2.000000	3846	21.639545	14.782900
D7-D8	43501	8175	2.000000	4387	24.810824	14.726516
D8-D9	49437	9514	2.000000	5125	114.699543	148.581810
D9-D10	55507	10448	2.000000	5689	8.977128	10.417522
D10-D11	60200	11560	2.000000	6325	11.601096	11.865162
D11-D12	62959	12218	2.000000	6667	4.842310	4.718830
D12-D13	70297	13768	2.321928	3658	10.290239	12.648093
D13-D14	78160	15851	2.321928	4241	14.898717	15.478102
D14-D15	83487	17011	2.321928	4590	8.628781	8.226856
D15-D16	86624	17720	2.321928	4781	3.836294	3.528541
D16-D17	94694	18515	2.321928	4991	5.188495	4.554895
D17-D18	101622	20138	2.321928	5420	10.607707	8.519013
D18-D19	104560	20698	2.321928	5575	3.792466	2.955694
D19-D20	112435	21898	2.321928	5922	9.228801	6.709938



(a) Saturation measurements **before** removing stop terms. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

(b) Saturation measurements **after** removing stop terms. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

Fig. 8: Visualization of saturation measurements on the RAW bags of terms extracted by NaCTeM TerMine

When looking at Fig. 8(a) and, especially, at 8(b), we observe that, as it was expected, the RAW collection is not terminologically saturated. Further, looking at the differences between Fig. 8 (a) and (b), we observe some nice indicators of the presence of noise in the textual documents of the collection. Indeed, the *thdr* values in Fig. 8(a) are much higher than the corresponding *thd* values. Though the *thd* values hint that the bags of terms might be close to saturation, the values of *thdr* are far beyond *eps*. Very interestingly, the values of *thd* measured after removing stop terms become very similar to that of *thdr*. At the same time the *thd* and *thdr* curves in Fig 8(b) very much resemble the *thdr* curve in Fig. 8(a). So, substantial differences between *thd* and *thdr* values signal about a possible need to clean the bags of terms, or the source texts, by removing the stop terms which have no relevance to the domain of the collection.

We then repeated the same experiment for the bags of terms extracted by the UPM Term Extractor. The results of measuring saturation look as follows.

The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) are presented in Table 9. These measurements are visualized in the diagrams of Fig. 9(a).

Table 9: Saturation measurements of the RAW bags of terms extracted by UPM Term Extractor

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
empty-D1	13385	5273	4.754888	1410	74.034379	100.000000
D1-D2	20796	8441	4.754888	2507	60.442609	44.946433
D2-D3	25127	10315	4.754888	3167	41.001680	23.448492
D3-D4	32128	13167	5.000000	2323	47.280334	29.444312
D4-D5	44506	17697	6.000000	3101	77.017676	42.471611
D5-D6	52569	21126	7.000000	3195	52.485783	31.362030
D6-D7	57681	23447	8.000000	3522	36.868580	22.400181
D7-D8	63920	26544	8.000000	4077	37.087374	21.495284
D8-D9	72242	30403	8.000000	4851	45.910520	26.133098
D9-D10	79915	33052	8.000000	5420	35.642461	20.960989
D10-D11	86238	35913	8.000000	6002	33.813132	20.581991
D11-D12	90121	37867	8.000000	6291	11.970775	7.312373
D12-D13	98735	41704	8.000000	6970	24.379570	14.527568
D13-D14	110737	47892	8.000000	7947	33.396561	17.943072
D14-D15	117491	51006	8.000000	8517	19.407655	10.327044
D15-D16	121296	52919	8.000000	8827	8.241292	4.310817
D16-D17	127956	55265	8.000000	9183	11.068328	5.638612
D17-D18	137195	59697	8.500000	6801	31.392994	19.553741
D18-D19	140346	61107	9.000000	6854	7.510382	4.738277
D19-D20	146538	64245	9.000000	7241	10.153605	6.130132

We then analyzed *B*20, extracted by UPM Extractor, going from the top of the list down to the terms having *c-value*s greater than 40. Based on this scan, we extracted the list of circa 220 stop terms. These stop terms have been removed from the bags of terms *B*1, …, *B*20 and saturation analysis has been repeated then. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) for so de-noised bags of terms are presented in Table 10. These measurements are visualized in the diagram of Fig. 9(b).

Table 10: Saturation measurements of the RAW bags of terms, extracted by UPM Term Extractor, after removing stop terms

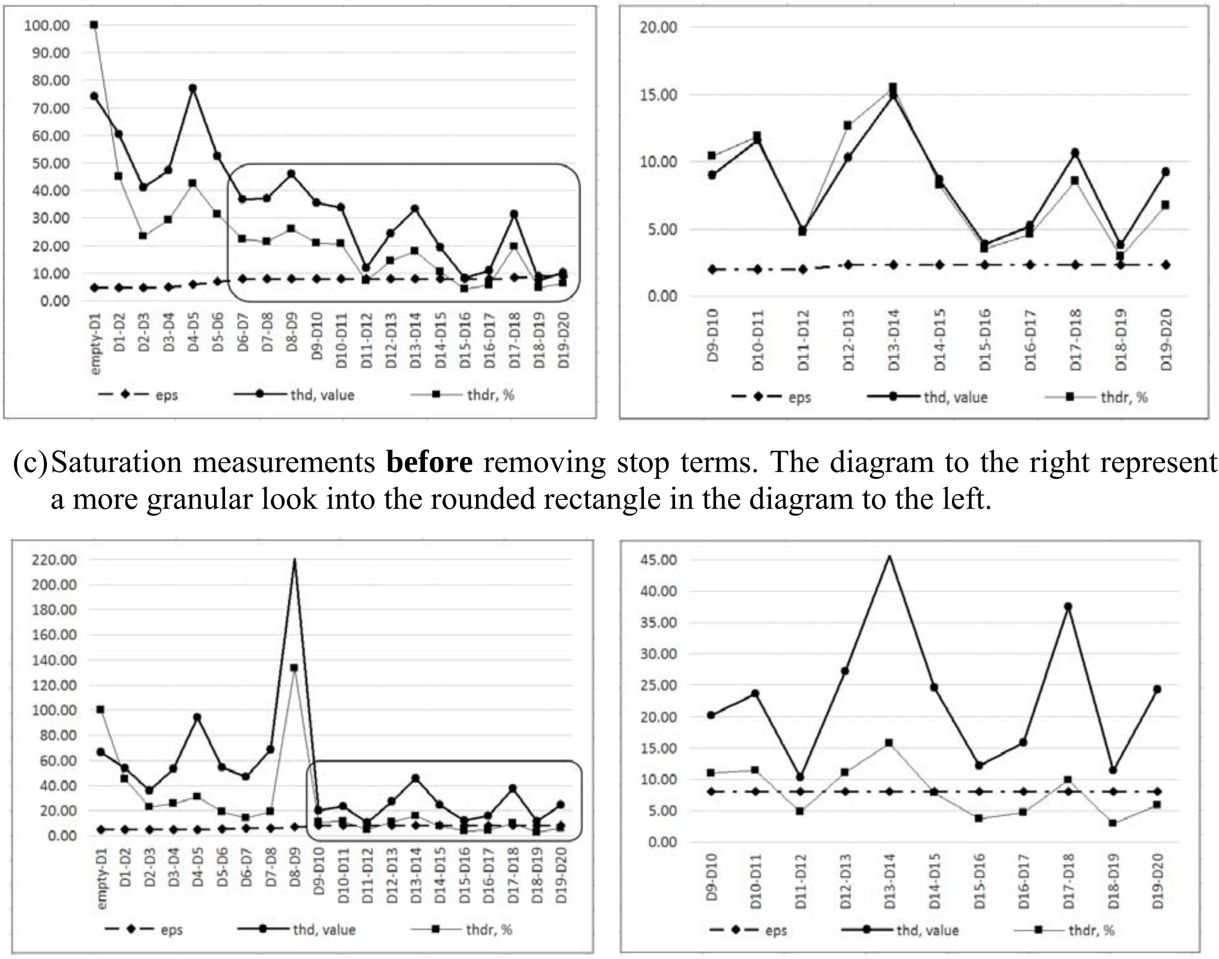
Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
empty-D1	13268	5172	4.754888	1350	66.277350	100.000000
D1-D2	20668	8324	4.754888	2413	54.139570	44.960102
D2-D3	24992	10192	4.754888	3057	35.698374	22.954975
D3-D4	31989	13039	4.754888	4030	53.242033	25.504336
D4-D5	44347	17545	4.754888	5655	94.468567	31.154574

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
D5-D6	52404	20965	5.500000	3703	54.530696	19.330335
D6-D7	57513	23282	6.000000	4102	46.700778	14.278223
D7-D8	63745	26371	6.000000	4753	68.665612	18.935646
D8-D9	72059	30222	7.000000	4847	220.786417	133.296062
D9-D10	79724	32862	8.000000	5233	20.162875	10.967976
D10-D11	86043	35719	8.000000	5809	23.678673	11.410709
D11-D12	89926	37673	8.000000	6098	10.276794	4.718681
D12-D13	98537	41507	8.000000	6774	27.214136	11.107645
D13-D14	110537	47693	8.000000	7748	45.657478	15.708147
D14-D15	117289	50805	8.000000	8316	24.584622	7.798558
D15-D16	121093	52717	8.000000	8625	12.147899	3.710487
D16-D17	127743	55053	8.000000	8971	15.860767	4.620703
D17-D18	136975	59478	8.000000	9823	37.518047	9.853141
D18-D19	140126	60888	8.000000	10087	11.360918	2.897207
D19-D20	146317	64024	8.000000	10633	24.335767	5.843354

Compared to the saturation measurements for the bags of terms extracted by TerMine, the values of *thd* for the bags of terms extracted by UPM Extractor form a clearer picture of the absence of saturation. In fact, the *thd* values measured on UPM Extractor results before removing the stop terms are 2.5-3 times higher than those measured on TerMine results after removing the stop terms. So, the results by UPM Extractor are more highly contrast compared to those of TerMine in terms of detecting the absence of saturation.

From the other hand, the values of *thdr* measured on TerMine results are a much sharper indicator of the need to de-noise the bags of terms. The *thdr* values measured on the UPM Extractor results do not differ from the corresponding *thd* values. If UPM Extractor is used to detect the absence of saturation, there is no real need however to analyze if *thdr* values indicate the presence of noise. So, overall the use of UPM Extractor is preferred in this case as it is a more precise instrument.

For this collection we did not measure if both tools extract statistically similar bags of terms in terms of terminological difference (*eps*, *thd*). This measurement would have no value for a case in which saturation is absent.



(c) Saturation measurements **before** removing stop terms. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

(d) Saturation measurements **after** removing stop terms. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

Fig. 9: Visualization of saturation measurements on the RAW bags of terms extracted by UPM Term Extractor

7.2 Terminological Saturation in Real Collections

We now present and analyze our results on measuring terminological saturation in our real document collections.

For the datasets extracted from the DMKD document collection the results look as follows.

We first processed at the bags of terms extracted by TerMine. The results of measuring individual term significant thresholds (*eps*) and terminological differences (*thd*, *thdr*) are presented in the saturation measurement analysis Table 11. These measurements are visualized in the diagrams of Fig. 10. The diagram at the left visualizes the entire table. The rounded rectangular circumscribes the area in the diagram at the left, which is presented in finer detail in the diagram at the right. The dashed vertical line points to the bag of terms (extracted from *D*14) in which saturation indicator has been observed for the first time as *thd* went below *eps*. The values of *eps*, no of retained terms, *thd*, and *thdr* for this bag of terms are bolded in Table 11.

The analysis of these results points out that there is a trend to reaching terminological saturation, perhaps for bigger datasets. The *eps* values have the tendency to go up and *thd*, *thdr* values go down with the increase in dataset numbers. The increase in the numbers of retained terms is also going down. There are three terminological peaks in the area of our closer interest at *D*10-*D*11, *D*12-*D*13, and *D*14-*D*15. The contribution of these peaks is not very significant however as the *thd* value increases not very much versus the vicinity – please see DAC results for comparison. Overall, it is too early to consider the DMKD collection saturated based on the extraction results by TerMine.

Table 11: Saturation measurements for the DMKD bags of terms extracted by NaCTeM TerMine

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
empty-D1	17820	4515	4.754888	608	25.215119	100.000000
D1-D2	37575	10113	5.000000	1352	24.974351	66.388195
D2-D3	51142	14598	6.000000	1655	15.251233	35.348047
D3-D4	70862	19816	6.339850	2023	16.067754	32.396313
D4-D5	88880	25069	6.666667	2406	13.366195	23.946410
D5-D6	103194	29511	7.000000	2767	10.651323	17.966432
D6-D7	116756	33939	7.250000	3008	11.304403	18.248761
D7-D8	130570	38314	7.924812	3312	9.384140	14.372626
D8-D9	145090	42238	7.924812	3611	7.393788	10.943197
D9-D10	158061	46070	8.000000	3835	8.775937	12.572770
D10-D11	185291	51883	9.000000	4014	14.936408	20.413596
D11-D12	197992	55872	9.000000	4371	6.409858	8.303101
D12-D13	226090	60516	9.000000	4700	9.513768	12.665386
D13-D14	241041	64392	**9.000000**	**5009**	**6.566537**	**8.604868**
D14-D15	256418	69067	9.000000	5438	7.828089	9.925786

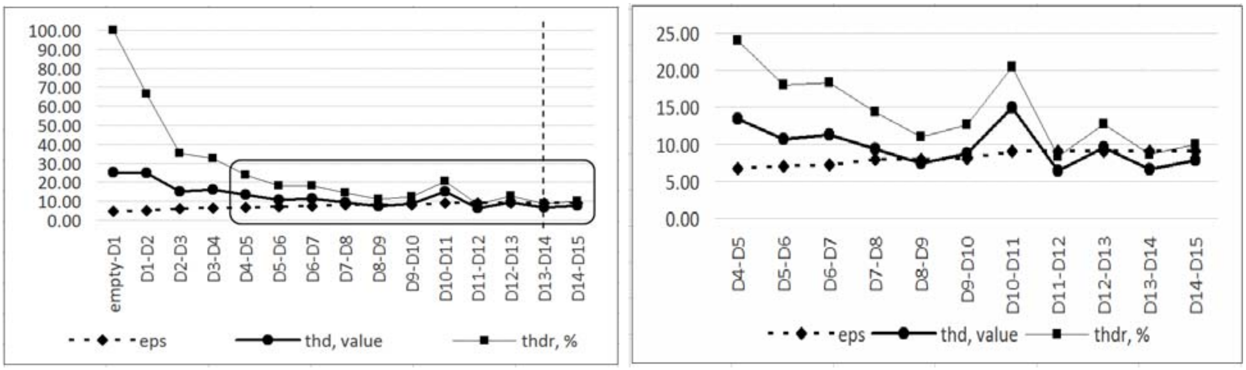


Fig. 10: Saturation measurements on the DMKD datasets based on the bags of terms extracted by NaCTeM TerMine. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

The results of measuring saturation based on the bags of terms extracted by UPM Term Extractor are presented in a numeric form in Table 12 and pictured diagrammatically in Fig. 11. Both Table 12 and Fig. 11 have the same structure as table 10 and Fig. 10 respectively: the measured values are in the table and the diagrams visualizing these values are in the figure. It could be noted that stable saturation is reached at *D*5-*D*6. The number of retained terms (from *B*6) is 4113, which is substantially lower than 5009 at the first potential saturation point in the TerMine case. Interestingly, *thd* and *thdr* values measured on UPM Term Extractor results behave quite similarly to those measured on TerMine results, also hinting about terminological peaks at the same points. The numbers of retained terms are lower, though not significantly, for UPM Term Extractor results. Saturation is reached due to much higher values of individual term significance threshold *eps*. Hence, for this document collection, **UPM Term Extractor** yields **better circumscribed** and **more compact** sets of **significant terms** and the cut-off happens for much higher values of term significance (*n-score*).

Table 12: Saturation measurements for the DMKD bags of terms extracted by UPM Term Extractor

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	23957	12891	9.509775	1480	35.883725	100.000000
D1-D2	49334	26461	14.000000	2212	27.697689	57.555144
D2-D3	67913	37202	15.500000	2770	17.059504	31.681566
D3-D4	89617	49323	17.000000	3242	16.200763	27.807386
D4-D5	112286	60971	19.019550	3770	13.225289	21.372369
D5-D6	130147	71007	**20.000000**	**4113**	**10.387994**	**15.724265**
D6-D7	147162	80333	22.000000	4448	9.996217	14.058018
D7-D8	164007	89635	23.774438	4666	8.861998	12.037252
D8-D9	182192	98866	24.000000	5190	9.451813	12.138658
D9-D10	200840	108760	26.500000	4986	9.420845	12.391220
D10-D11	230283	122406	28.529325	5709	12.048718	15.206691
D11-D12	250739	133418	36.000000	4825	15.574182	17.565133
D12-D13	275270	145576	36.000000	5285	8.591505	9.532408
D13-D14	298786	156733	37.000000	5503	6.539858	6.987360
D14-D15	320025	167888	38.039100	5800	7.536281	7.726895

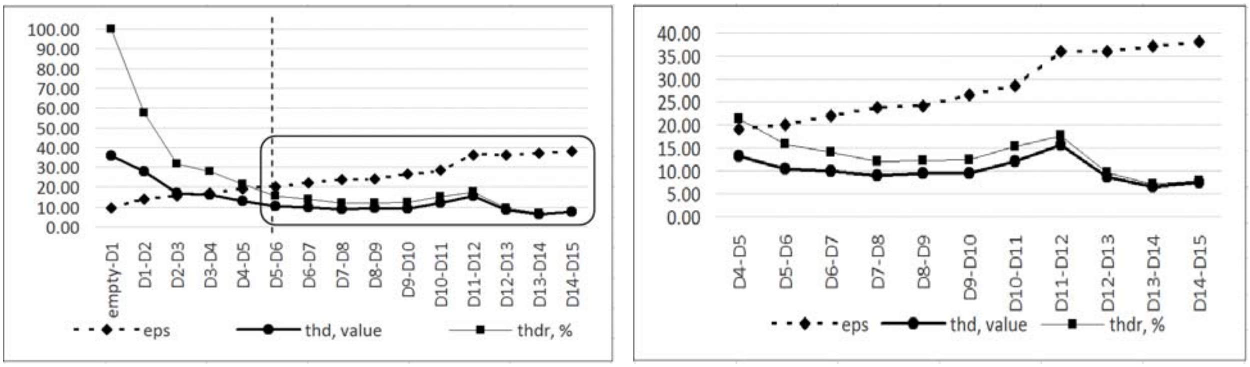


Fig. 11: Saturation measurements on the DMKD datasets based on the bags of terms extracted by UPM Term Extractor. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

One hypothesis about the reason for better UPM Term Extractor performance could be that it extracts not all the terms from the documents it takes in, and NaCTeM TerMine reaches substantially higher recall values. To check that, we measured terminological differences between the bags of terms extracted, from the same datasets by UPM Extractor and TerMine. The result is presented in a numeric form in Table 13 and pictured diagrammatically in Fig. 12.

Table 13: Comparison of the retained sets of terms extracted from DMKD collection by UPM Term Extractor and NaCTeM TerMine

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
D1-D1m	34415	15561	8.000000	1802	10.388350	29.985197
D2-D2m	71464	32481	9.509775	2924	24.827366	40.289915
D3-D3m	97594	45806	10.000000	3885	30.966042	41.735911
D4-D4m	130712	60874	12.000000	4328	29.383461	38.755006
D5-D5m	161876	75499	13.168361	4792	29.817941	38.364136
D6-D6m	187072	87912	**14.000000**	5556	**25.137306**	**33.246112**
D7-D7m	211850	99820	15.333333	5541	24.538215	31.158022
D8-D8m	236823	111615	16.000000	5940	23.152435	29.830633
D9-D9m	263161	122936	16.000000	6692	21.916281	28.319989
D10-D10m	288548	134865	17.500000	6592	18.511344	25.745032
D11-D11m	330100	151481	**19.019550**	7193	**18.635161**	**24.974531**
D12-D12m	356679	164498	19.019550	7825	29.677258	37.755228
D13-D13m	399383	179017	19.651484	8223	28.553048	35.997855
D14-D14m	430018	192170	20.000000	8595	27.450743	33.963834
D15-D15m	459589	205837	21.333334	8768	27.054264	32.436521

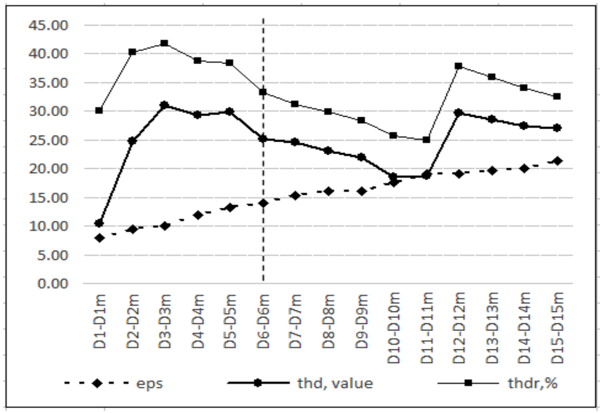


Fig. 12: Comparison of the retained sets of terms extracted from the DMKD collection by UPM Term Extractor and NaCTeM TerMine

Overall, Table 13 and Fig. 12 show that both tools extract somewhat similar bags of terms. This similarity increases with the growth of a dataset and the individual term significance thresholds (*eps*) are similar in values to the case of UPM Term Extractor. The numbers of retained terms are higher, however, than in Fig. 10 and 11. These also hint that the extracted bags of terms are similar and recall values of individual tools are similar and acceptable.

Interestingly, terminological difference (*thd*) in Fig. 12 goes below *eps* exactly at the point when TerMine results show the highest terminological peak (c.f. Fig. 10). So, it looks like both tools extract similar bags of terms but TerMine reaches the saturation level a bit later, when it collects the contribution from the increment at the highest terminology peak. Yet interestingly, *thd* values go beyond *eps* after *D*11. We think[25] that the reason for that is the increasing influence of the accumulated noise in the datasets, which is processed differently by the individual tools.

The results of saturation measurements for the TIME document collection are presented in a numeric form in Tables 14-16 and pictured in the diagrams presented in Fig. 13-15. These diagrams, and also the diagrams for the DAC collection in Fig. 16-18, are built similarly to the diagrams in Fig. 10-12.

Table 14: Saturation measurements for the TIME bags of terms extracted by NaCTeM TerMine

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
empty-D1	11593	1888	2.807355	511	55.241461	100.000000
D1-D2	19853	3580	3.000000	932	48.567987	63.996536
D2-D3	30197	6407	3.000000	1565	55.020634	52.678543
D3-D4	41602	9451	3.000000	2384	60.292173	52.973052

[25] We did not yet check this. So, it is only a hypothesis.

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
D4-D5	53527	12755	3.500000	2171	56.627975	56.610404
D5-D6	65862	16214	4.000000	2718	36.047962	30.067689
D6-D7	84219	19763	4.000000	3189	28.817533	24.099603
D7-D8	91455	22159	4.000000	3687	27.084555	22.411852
D8-D9	103340	25015	4.000000	4154	22.526122	18.538041
D9-D10	118966	28437	4.000000	4615	23.079828	19.075832
D10-D11	131417	31207	4.000000	5068	39.754947	40.636472
D11-D12	148354	34938	4.000000	5531	13.658011	13.668833
D12-D13	156376	37293	4.000000	5919	15.185382	15.770276
D13-D14	171978	41133	4.000000	6463	12.767795	13.100688
D14-D15	181803	44153	4.000000	6988	10.432455	10.354328
D15-D16	193764	47315	4.000000	7523	12.676934	12.317197
D16-D17	208257	51011	4.378492	6551	11.964115	12.414309
D17-D18	236549	55759	4.754888	7107	17.062060	17.367168
D18-D19	247255	58684	4.754888	7612	13.404024	13.987011
D19-D20	262821	62582	5.000000	7369	14.933883	17.061322
D20-D21	277630	66343	5.000000	7928	10.715013	12.382464
D21-D22	287804	69321	5.000000	8343	7.401955	8.540399

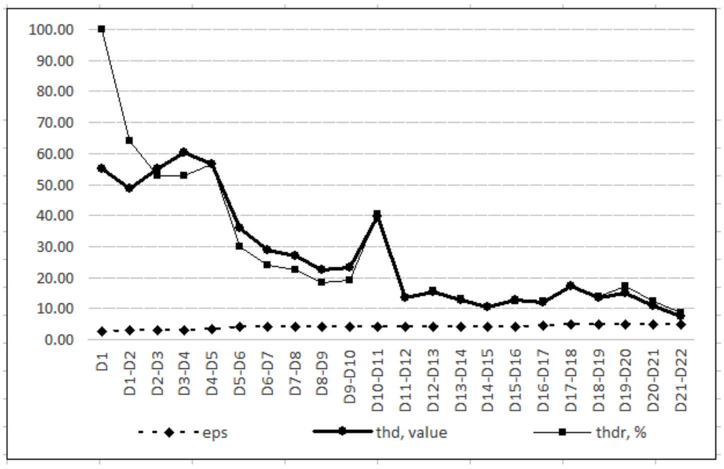


Fig. 13: Saturation measurements on the TIME datasets based on the bags of terms extracted by NaCTeM TerMine.

The saturation measurements based on the bags of terms extracted by TerMine **did not show any saturation** – as pictured in Fig. 13. The *thd* values did not go below *eps*. The tendency is similar to the DMKD experiment however: a trend to reaching terminological saturation, perhaps for bigger datasets. The *eps* values go up with the increase in dataset numbers, though significantly slower than in the DMKD case. The maximal observed *eps* value is 5 for TIME versus 9 for DMKD. The *thd* and *thdr* values go down with the increase in dataset numbers, but not quickly enough to go below *eps*. As a consequence, the maximal number of retained terms is significantly higher that in the DMKD case: 8343 versus 5438, though the difference in the extracted numbers of terms is not that significant:

~287K versus ~253K. Interestingly, the terminological peaks in the TIME collection are observed at *D*3-*D*4, *D*10-*D*11, *D*17-*D*18, and *D*19-*D*20. The highest peak is at *D*10-*D*11, which repeats the DMKD case, probably by a coincidence. Similarly to DMKD, the contribution of these peaks is not very substantial as the *thd* value increases not very much compared to the neighbourhood.

Table 15: Saturation measurements for the TIME bags of terms extracted by UPM Term Extractor

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
empty-D1	53478	13775	28.000000	1379	112.240776	100.000000
D1-D2	91701	23816	24.000000	2473	72.425797	59.389624
D2-D3	114061	32419	21.500000	3028	24.265441	17.312132
D3-D4	129896	39643	19.651484	3997	32.879384	20.295700
D4-D5	145796	46702	19.651484	4466	32.622249	17.809632
D5-D6	162746	54629	20.000000	4587	44.646245	27.027091
D6-D7	190263	63684	21.000000	5133	38.071510	24.076680
D7-D8	200176	69097	22.000000	5413	26.869088	18.598430
D8-D9	217461	76315	22.000000	5855	18.776156	13.110501
D9-D10	245967	84664	23.219281	6453	26.914239	18.281013
D10-D11	263034	91132	24.000000	6428	24.164533	16.688847
D11-D12	287887	99231	**23.774438**	**7110**	**18.109566**	**12.737127**
D12-D13	298367	104398	23.774438	7383	12.573733	9.144105
D13-D14	320500	112898	24.000000	7723	13.334954	9.624406
D14-D15	333975	119787	23.774438	8298	14.403930	10.698614
D15-D16	350741	127257	24.000000	8426	16.428110	13.135633
D16-D17	369316	135085	24.000000	8877	9.642629	7.638542
D17-D18	389022	143452	24.000000	9617	11.416546	8.784302
D18-D19	399553	148896	24.000000	10005	8.042102	6.136623
D19-D20	420464	158179	24.000000	10574	11.655716	8.652365
D20-D21	435075	165519	26.000000	9751	9.781677	7.297311
D21-D22	449719	171135	26.000000	10139	6.926144	5.109224

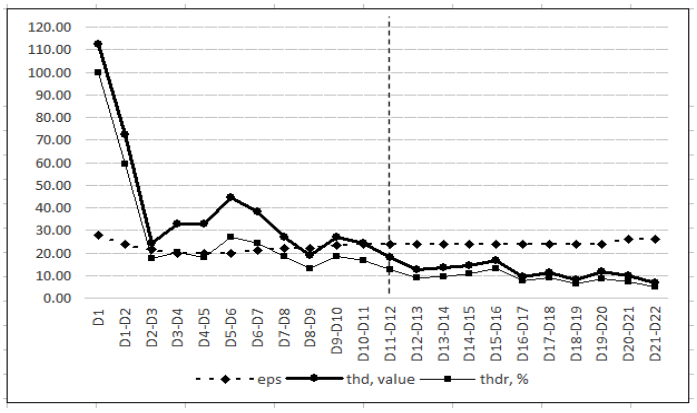


Fig. 14: Saturation measurements on the TIME datasets based on the bags of terms extracted by UPM Term Extractor.

The saturation measurements based on the bags of terms extracted by UPM Term Extractor **reveal stable saturation** starting from *D*11-*D*12 – as presented in Table 15 by bolded values and pictured in Fig. 14 by the vertical dashed line. The values of *thd* and *thdr* resemble these of the TerMine case, so the saturation curve has terminological peaks nearly at the same points. The height of those peaks is however lower. The values of individual term significance threshold *eps* are however much higher – similarly to the DMKD experiment. Saturation is detected at *eps* equal to 23.774, whereas the values of *eps* in the TerMine case do not increase beyond 5.000. The number of retained terms (from *B*12) is 7110, which is only 2.47% of the total number of extracted terms in the corresponding bag of terms *B*12. Therefore, we may draw a similar conclusion for this experiment. Saturation is reached due to much higher values of individual term significance threshold *eps*. For the TIME document collection, **UPM Term Extractor** yields **better circumscribed** and **more compact** sets of **significant terms** and the cut-off happens for much higher values of term significance (*n-score*).

Table 16: Comparison of the retained sets of terms extracted from TIME collection by UPM Term Extractor and NaCTeM TerMine

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
D1-D1m	57869	14731	24.000000	1517	14.655056	12.800024
D2-D2m	99710	25730	23.219281	2389	8.504236	7.505038
D3-D3m	126208	35665	19.651484	3216	9.720847	7.085422
D4-D4m	147165	44375	18.000000	3795	9.150792	6.303159
D5-D5m	168294	53015	16.000000	4599	21.309961	13.169097
D6-D6m	191106	62718	16.000000	5155	19.956569	13.420014
D7-D7m	227790	73884	16.000000	5998	19.061987	13.134541
D8-D8m	241040	80499	16.000000	6346	18.103268	13.745978
D9-D9m	264760	89394	16.000000	6816	17.654014	13.573892
D10-D10m	300701	99660	**17.777779**	7045	**18.502197**	**14.037191**
D11-D11m	323715	107769	17.000000	7562	26.566239	20.949834
D12-D12m	356448	117924	**16.000000**	8846	**29.109029**	**22.977325**
D13-D13m	371180	124442	16.000000	9181	27.402389	22.414044
D14-D14m	400533	135102	16.000000	9844	28.088519	22.503837
D15-D15m	418519	143498	16.000000	10212	25.689242	21.407490
D16-D16m	441618	152802	16.000000	10695	25.112180	22.354416
D17-D17m	466902	162606	16.000000	11196	25.833841	22.971737
D18-D18m	504661	173658	16.000000	11844	26.884985	23.607541
D19-D19m	521629	180818	16.000000	12376	28.185740	24.531213
D20-D20m	550288	192227	16.500000	12037	27.873027	24.379546
D21-D21m	573929	201594	18.000000	12308	28.122244	24.044072
D22-D22m	595003	209089	18.110527	12483	27.240609	23.164606

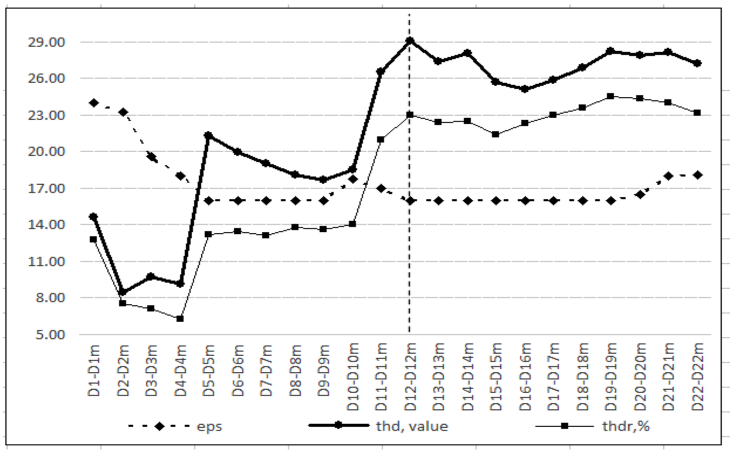


Fig. 15: Comparison of the retained sets of terms extracted from the TIME collection by UPM Term Extractor and NaCTeM TerMine

We also checked if both tools extract similar bags of terms from the TIME collection. The results have been measured following the same approach as in the case of DMKD (Table 12) and are pictured in Fig. 15. It could be seen from the figure, that the terminological difference (*thd*) between the bags of retained terms at the saturation point *D*12-*D*12*m*[26] equals to ~29, while *eps* equals to 16. So, *thd* is 1.81 times higher than *eps*. In the DMKD case the difference between *thd* and *eps* at the saturation point is slightly lower – 1.80 times. Very similarly to the DMKD case, the difference grows after the saturation point, which, as we believe, could be explained by the same reason – the influence of the accumulated noise in the datasets beyond the saturation point. Hence, manual cleaning of the TIME datasets did not really help a lot, as the results very much resemble the DMKD case, for which the datasets were not cleaned.

The results of saturation measurements for the DAC document collection are presented in a numeric form in Tables 17-19 and pictured in Fig. 16-18. By its design, the DAC document collection appears to be much noisier than DMKD and TIME. The results also differ – in values but not in the overall picture.

Table 17: Saturation measurements for the DAC bags of terms extracted by NaCTeM TerMine

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	21414	3731	3.000000	907	87.884036	100.000000
D1-D2	36971	7022	3.000000	1784	85.921132	66.191975
D2-D3	60192	11521	3.000000	2834	78.885421	41.010879

[26] *D*12 is the dataset from which *B*12 is extracted by UPM Extractor and *B*12*m* by TerMine. *B*12*m* is further converted to the UPM Extractor format and the pair (*B*12, *B*12*m*) is fed into the THD module. The module returns *eps*, *thd*, and *thdr* values for the pair as described in Section 3.

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
D3-D4	82160	15064	3.000000	3743	68.212110	27.753447
D4-D5	94869	18447	3.169925	3887	92.527503	40.955982
D5-D6	116629	22772	3.169925	4754	82.812749	39.627153
D6-D7	130044	26196	3.169925	5568	43.381294	17.745804
D7-D8	142670	29403	3.321928	5152	82.007135	46.948761
D8-D9	154252	32634	3.500000	5543	26.066742	14.354607
D9-D10	179659	37792	3.459432	6425	35.018676	18.160732
D10-D11	194091	41045	3.807355	6618	28.992512	15.073077
D11-D12	213845	45219	3.906891	7256	28.169761	14.206589
D12-D13	235205	49236	3.906891	7832	22.148436	10.787918
D13-D14	249312	53148	4.000000	8302	22.937860	10.760466
D14-D15	265265	56871	4.000000	8887	17.735412	7.910243
D15-D16	281291	60338	4.000000	9573	24.757868	10.679613
D16-D17	299288	64413	4.000000	10174	36.133490	16.851960
D17-D18	325758	69724	4.000000	10908	19.173440	8.446286
D18-D19	340694	73351	4.000000	11537	16.351450	6.783656
D19-D20	361091	77559	4.000000	12159	22.925510	9.573753
D20-D21	382193	81912	4.000000	12774	20.048766	8.315987
D21-D22	398044	85547	4.000000	13439	19.364574	7.992865
D22-D23	422011	90520	4.000000	14047	14.028586	5.626976
D23-D24	437137	94374	4.000000	14712	21.506053	8.743225
D24-D25	480360	98766	6.333333	8105	135.489690	87.770152
D25-D26	489016	100077	6.207769	8261	1.632171	1.046258

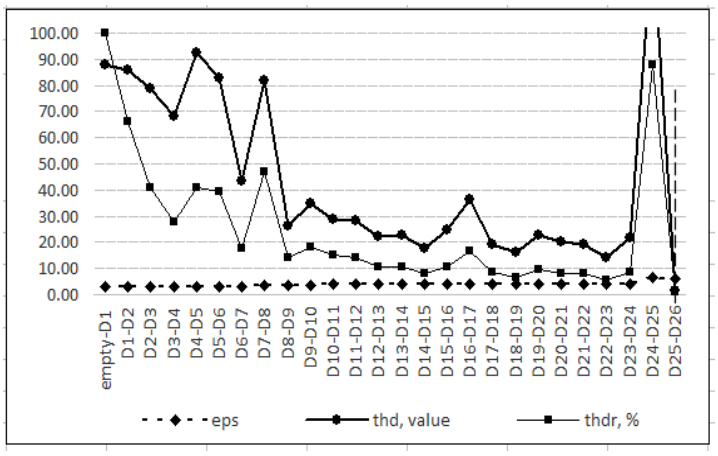


Fig. 16: Saturation measurements on the DAC datasets based on the bags of terms extracted by NaCTeM TerMine.

The saturation measurements based on the bags of terms extracted by TerMine revealed the potential saturation point only in the last measurement at *D*25-*D*26 – as pictured in Fig. 16. However, the terminological peak at *D*24-*D*25, with *thd* equal to 135.49, hints about the further instability. So, speaking about a tendency to reach stable saturation later would be a speculation. More measurements are needed to judge about it.

It is also interesting to compare the saturation behaviour in DAC to that in TIME, as both collections come from the same publisher, so have the same layout, and represent papers of similar size. The difference is that TIME was manually cleaned and DAC was not. Tables 14 and 17 show the differences in measured values for the datasets of roughly similar sizes.

The comparison of the measurements for TIME and DAC, based on the extraction results by TerMine, reveals that:

- The values of eps grow faster for TIME than for DAC
- The numbers of extracted and retained terms for DAC are substantially higher than for TIME
- The numbers of retained terms for TIME grow monotonically and this growth slows down – which is an indicator of possible saturation in the upcoming measurements. The picture is different for DAC. The number of retained terms substantially drops below the previous value at D24-D25 and the *thd* dramatically picks up from 21.51 to 135.49. Further, the process seems to start recovering with the number of retained terms going slightly up and the thd dropping down to 1.63.

We believe, again, that the reason for the peak at *D*24-*D*25 is the influence of the accumulated noise. So, cleaning the TIME collection still helped to have a more stable saturation process. However, TerMine results signal about a problem with data quite lately in the process.

Table 18: Saturation measurements for the DAC bags of terms extracted by UPM Term Extractor

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
empty-D1	27395	9648	9.509775	1529	144.244459	100.000000
D1-D2	46574	17773	11.609640	2344	131.870673	52.541377
D2-D3	77317	28133	11.609640	3698	160.895398	73.734239
D3-D4	96012	35834	11.609640	4747	71.404424	30.800104
D4-D5	112551	43133	24.000000	2080	154.266320	225.375923
D5-D6	138766	51942	21.000000	2848	12.757374	15.709925
D6-D7	156527	59096	36.000000	1661	19.087158	35.414789
D7-D8	169982	65725	33.219281	2107	4.975734	8.451820
D8-D9	184272	72510	32.000000	2471	4.523149	7.134879
D9-D10	212542	82279	28.529325	3510	8.988935	12.418430
D10-D11	230726	89000	18294.037199	34	1.605671	5.624197
D11-D12	256595	97227	16058.681574	37	1.463132	4.875078
D12-D13	281606	105161	13940.066402	39	0.828380	2.685982
D13-D14	301187	113103	11712.000000	41	1.268232	4.045308
D14-D15	321240	120405	17240.806813	39	0.729627	2.354560
D15-D16	337402	126847	15149.071582	41	0.757379	2.385808

Datasets Pair	Bag of Terms	C-value > 1	eps	Retained Terms (C-value > eps)	thd, value	thdr,%
D16-D17	357543	134022	10985.179810	44	0.908878	2.783354
D17-D18	386999	143030	6814.858958	48	0.830762	2.481010
D18-D19	406035	150172	2352.000000	56	0.664386	1.945539
D19-D20	427894	158040	1074.000000	78	0.895210	2.554499
D20-D21	453189	166539	710.000000	128	1.100893	3.045742
D21-D22	470374	173179	567.000000	175	0.790483	2.140153
D22-D23	497532	182472	464.000000	246	1.006809	2.653504
D23-D24	515285	189861	454.000000	262	0.821752	2.168438
D24-D25	543322	195944	398.000000	319	0.731177	1.892906
D25-D26	552077	198416	376.000000	346	0.279283	0.717830

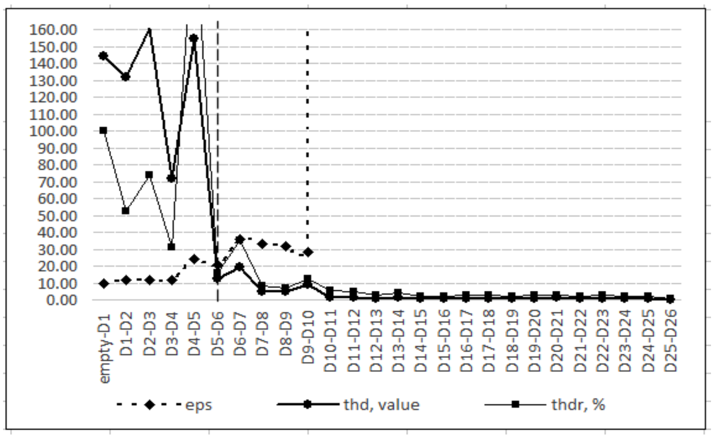


Fig. 17: Saturation measurements on the DAC datasets based on the bags of terms extracted by UPM Term Extractor.

The saturation measurements based on the bags of terms extracted by UPM Term Extractor **reveal stable saturation** starting from *D*5-*D*6 with *eps* at about 20 – as pictured in Fig. 17 by the vertical dashed line. However, as it is seen in the figure and also in Table 18, the values of *eps* peak up to 18 294 at *D*10-*D*11 and the numbers of retained terms go down to 34 which is more than 100 times less than the previous value. A closer examination of the bags of terms revealed that these 34 terms are nothing but the noise which has been accumulated much earlier in the case the use of UPM Extractor. Therefore, in the case of a noisy document collection, UPM Extractor is much more sensitive in detecting excessive noise, compared to TerMine. So, the situation pictured in Table 18 could be used as an indicator of the need to clean the collection before terminology extraction. Otherwise the result will be of zero quality.

Though not very relevant for this collection, we still compared if the bags of terms extracted by both tools were statistically similar. The result is presented in Table 19 and pictured in Fig. 18. The comparison showed that, starting from *D*5, where *thd* equals to 3.97 and *eps* to 19.65, both tools successfully extracted the very similar sets of accumulated noise terms.

Table 19: Comparison of the retained sets of terms extracted from DAC collection by UPM Term Extractor and NaCTeM TerMine

Datasets Pair	**Bag of Terms**	**C-value > 1**	**eps**	**Retained Terms (C-value > eps)**	**thd, value**	**thdr,%**
D1-D1m	37890	11649	8.000000	1769	35.484411	27.436853
D2-D2m	65098	21545	8.000000	3269	63.038225	25.307750
D3-D3m	106217	34278	8.000000	5104	57.257327	27.023713
D4-D4m	139531	43897	8.000000	6629	61.401730	27.000544
D5-D5m	161884	52870	19.651484	2133	3.966033	6.135380
D6-D6m	198167	63978	18.500000	2808	4.685435	6.216670
D7-D7m	222146	72776	34.500000	1377	1.132364	2.255087
D8-D8m	241947	81004	31.500000	1770	1.474268	2.707691
D9-D9m	261834	89432	28.529325	2272	1.862434	3.154840
D10-D10m	303215	102030	26.000000	2974	2.563050	3.899405
D11-D11m	328476	110466	20264.937926	32	0.000000	0.000000
D12-D12m	363225	120941	18294.037199	34	0.000000	0.000000
D13-D13m	398745	131132	16186.071914	36	0.000000	0.000000
D14-D14m	425162	141012	15413.746360	38	0.000000	0.000000
D15-D15m	452439	150222	20885.254082	36	0.000000	0.000000
D16-D16m	476492	158432	18497.395802	38	0.000000	0.000000
D17-D17m	505409	167763	17240.806813	39	0.000000	0.000000
D18-D18m	548185	179822	13263.062046	42	0.000000	0.000000
D19-D19m	573857	188651	10985.179810	44	0.000000	0.000000
D20-D20m	606075	198913	8003.417364	47	0.000000	0.000000
D21-D21m	641497	209750	2671.000000	55	0.000000	0.000000
D22-D22m	665857	218158	1400.500000	70	0.000000	0.000000
D23-D23m	705073	230251	873.500000	103	0.000000	0.000000
D24-D24m	731084	239737	864.665316	111	0.000000	0.000000
D25-D25m	779648	248693	658.000000	174	0.643133	1.733769
D26-D26m	792579	251853	624.000000	188	0.016050	0.043002

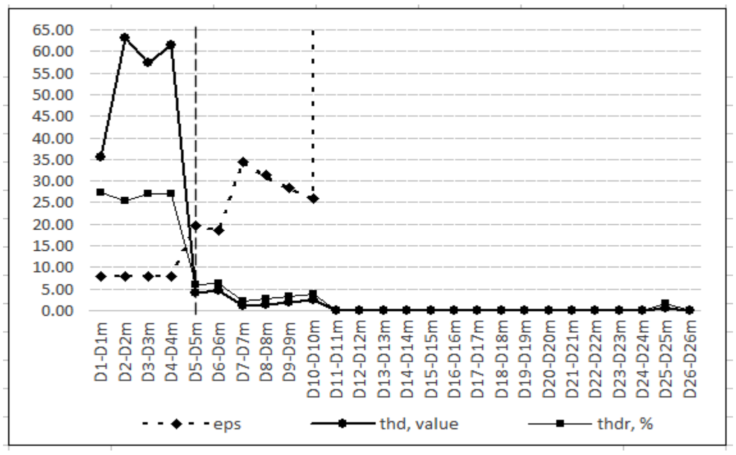


Fig. 18: Comparison of the retained sets of terms extracted from the DAC collection by UPM Term Extractor and NaCTeM TerMine

Finally, we are glad to note that *thd* value peaks in the DAC case did not occur at the same measurement points as for TIME or DMKD. So, it may be believed that these peaks are not caused by the internal workings of the term extraction method and its implementation in a tool. These indeed signal about the increased terminological contribution by the corresponding dataset.

8 Conclusions and Recommendations

This section summarizes our findings after analyzing the results of the experiments on cross-evaluating NaCTeM TerMine and UPM Term Extractor. The summary is structured along the cases based on our document collections.

8.1 Synthetic Collections

Case 1: 1DOC – quick saturation expected. For the bags of terms extracted by both tools very stable saturation has been observed quite quickly – which was expected. The differences in saturation measurements are as follows: (i) UPM Extractor generated bigger bags of terms with *c-value* > 1: 3 019 terms versus 1 208 in the TerMine case; (ii) individual term significance thresholds (*eps*) were about 2.5 times higher for UPM Extractor; (iii) the number of retained terms with *c-value* > *eps* was approximately 2 times bigger in the UPM Extractor case; (iv) the values of *thd* and *thdr* were significantly lower (~10 000 times) for TerMine. Overall, TerMine results showed a slightly quicker convergence to saturation than that by UPM Extractor. From the other hand: (i) the number of retained terms from the saturated sub-collection; and (ii) the cut-off point at the individual term significance threshold were higher in the UPM Extractor results. Both tools extracted statistically similar bags of terms despite the fact that the numbers of retained terms differed significantly. Overall, both tools behaved, in detecting saturation and extracting similar bags of terms, exactly as expected by the design of the case.

Conclusion (case 1): These results confirm the **adequacy of our saturation metric** for the boundary case of immediate saturation

Conclusion (case 1): Linguistically, **TerMine** is **more selective** in extracting **term candidates**. So, the pre-processing in TerMine is more sophisticated and, probably, **more accurate**. From the other hand, the **cut-offs** in **UPM Extractor** outputs happen for **substantially more significant terms**. So, the statistical processing part of UPM Extractor circumscribes more compact, yet more significant sets of terms. Hence, **UPM Extractor** is overall a **more precise** instrument.

Case 2: RAW – saturation should not be reached. While measuring saturation in the bags of terms extracted by TerMine, we observed that saturation has not been reached. We also noticed that the measurements of thd and thdr on these bags of terms differed noticeably for the cases before and after removing stop terms. So, these differences between *thd* and *thdr* values signal about a possible need to clean the bags of terms, or the source texts, by removing the stop terms which have no relevance to the domain of the collection. The *thd* values measured on UPM Extractor results before removing the stop terms are 2.5-3 times higher than those measured on TerMine results after removing the stop terms. So, the results by UPM Extractor are more highly contrast compared to those of TerMine in terms of detecting the absence of saturation. Overall, both tools behaved, in failing to detect saturation and extracting similar bags of terms, as expected by the design of the case.

Conclusion (case 2): **TerMine** is **more sensitive** in indicating the **need to de-noise** the bags of terms.

Conclusion (case 2): **UPM Extractor** is a **more precise** instrument to detect the **absence of saturation**

Conclusion (case 2): These results confirm the **adequacy of our saturation metric** for the boundary case for non-reachable saturation

Recommendation: The use of **UPM Extractor is preferred** to detect that saturation is hardly expected.

8.2 Real Collections

Case 3: DMKD collection (automatically pre-processed). Overall, it cannot be reliably judged that the DMKD collection is saturated based on the extraction results by TerMine (see Table 11 and Fig. 10). In difference to that, the saturation measurements using the bags of terms extracted by UPM Extractor show stable saturation quite quickly (see Table 12 and Fig. 11). For this document collection, **UPM Term Extractor** yields **better circumscribed** and **more compact** sets of **significant terms** and the cut-off happens for much higher values of term significance (*n-score*). It has also been noticed that both tools extracted statistically similar bags of terms in terms of terminological difference.

Case 4: TIME collection (manually cleaned). Saturation measurements using the bags of terms extracted by TerMine failed to detect saturation in the TIME collection (see Table 14 and Fig. 13). Very similarly to the DMKD case, the saturation measurements using the bags of terms extracted by UPM Extractor reveal stable saturation quite quickly (see Table 15 and Fig. 14), also with much higher individual term importance thresholds *eps*. These result in significantly more compact sets of retained significant terms (2.47% of all extracted). Hence, for this document collection, **UPM Term Extractor** also yields **better circumscribed** and **more compact** sets of **significant terms**.

Conclusion (cases 3 and 4): Both tools yielded similar results in detecting saturation and retaining significant terms for DMKD and TIME collections. **Manual cleaning** of the TIME collection did not help noticeably for improving the results of saturation measurements – therefore is **not** really **necessary**.

Case 5: DAC collection (very noisy). UPM Extractor demonstrated the capacity to accumulate excessive noise from the datasets to the bags of terms substantially earlier than TerMine. The saturation curve (see Fig. 17), built for the measurements using UPM Extractor results, signals about this noise quite sharply – with the numbers of retained significant terms dropping down by two orders of magnitude and individual term significance thresholds going up by three orders of magnitude.

Conclusion (case 5): In the case of noisy datasets and due to not being very selective in extracting term candidates, **UPM Extractor** is **much more sensitive** in detecting **excessive noise**, compared to TerMine.

Recommendation: The use of **UPM Extractor is preferred** over TerMine to detect **terminological saturation** or **excessive noise**; this is not constrained by a subject domain and does not depend on manual de-noising of the source data in the collection.

Acknowledgements

The first author is funded by a PhD grant provided by Zaporizhzhia National University and the Ministry of Education and Science of Ukraine.

The research leading to this report has been done in part in cooperation with the Ontology Engineering Group of the Universidad Politécnica de Madrid in frame of FP7 Marie Curie IRSES SemData project (http://www.semdata-project.eu/), grant agreement No PIRSES-GA-2013-612551.

A substantial part of the instrumental software used in the reported experiments has been developed in cooperation with BWT Group.

The data collection of Springer journal papers dealing with Knowledge Management, including DMKD, has been provided by Springer-Verlag.

References

1. Kosa, V., Chugunenko, A., Yuschenko, E., Badenes, C., Ermolayev, V., Birukou, A.: Semantic Saturation in Retrospective Text Document Collections. In: Mallet, F., Zholtkevych, G. (eds.) Proc. ICTERI 2017 PhD Symposium, CEUR-WS, vol. 1851, pp. 1--8, Kyiv, Ukraine, May 16-17 (2017) online
2. Tatarintseva, O., Ermolayev, V., Keller, B., Matzke, W.-E.: Quantifying Ontology Fitness in OntoElect Using Saturation- and Vote-Based Metrics. In: Ermolayev, V., et al. (eds.) Revised Selected Papers of ICTERI 2013, CCIS, vol. 412, pp. 136--162 (2013)
3. Osborne, F., Motta, E.: Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks. In: Arenas, M. et al. (eds.): ISWC 2015, Part I, LNCS, vol. 9366, pp. 408--424 (2015). doi: 10.1007/978-3-319-25007-6_24
4. Astrakhantsev, N.: ATR4S: Toolkit with State-of-the-art Automatic Terms Recognition Methods in Scala. arXiv preprint arXiv:1611.07804 (2016)
5. Zhang, Z., Iria, J., Brewster, C., Ciravegna, F.: A comparative evaluation of term recognition algorithms. In: Proc. Sixth Int Conf on Language Resources and Evaluation, LREC08, Marrakech, Morocco (2008)
6. Fahmi, I., Bouma, G., van der Plas, L.: Improving statistical method using known terms for automatic term extraction. In: Computational Linguistics in the Netherlands, CLIN 17 (2007)
7. Wermter, J., Hahn, U.: Finding new terminology in very large corpora. In: Clark, P., Schreiber, G. (eds.) Proc.3rd Int Conf on Knowledge Capture, K-CAP 2005, pp. 137--144, Banff, Alberta, Canada, ACM (2005) doi: 10.1145/1088622.1088648
8. Daille, B.: Study and implementation of combined techniques for automatic extraction of terminology. In: Klavans, J., Resnik, P. (eds.) The Balancing Act: Combining Symbolic and Statistical Approaches to Language, pp. 49--66. The MIT Press. Cambridge, Massachusetts (1996)
9. Cohen, J. D.: Highlights: Language- and domain-independent automatic indexing terms for abstracting. Journal of the American Society for Information Science 46(3), 162--174 (1995). doi: 10.1002/(SICI)1097-4571(199504)46:3<162::AID-ASI2>3.0.CO;2-6
10. Caraballo, S. A., Charniak, E.: Determining the specificity of nouns from text. In: Proc. 1999 Joint SIGDAT Conf on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63--70. (1999)
11. Medelyan, O., Witten, I. H.: Thesaurus based automatic keyphrase indexing. In: Marchionini, G., Nelson, M. L., Marshall, C. C. (eds.) Proc. ACM/IEEE Joint Conf on Digital Libraries, JCDL 2006, pp. 296--297, Chapel Hill, NC, USA, ACM (2006) doi: 10.1145/1141753.1141819
12. Ahmad, K., Gillam, L., Tostevin, L.: University of surrey participation in trec8: Weirdness indexing for logical document extrapolation and retrieval (wilder). In: Proc. 8th Text REtrieval Conf, TREC-8 (1999)
13. Frantzi, K. T., Ananiadou, S.: The c/nc value domain independent method for multi-word term extraction. Journal of Natural Language Processing 6(3), 145--180 (1999). doi: 10.5715/jnlp.6.3_145
14. Sclano, F., Velardi, P.: TermExtractor: A Web application to learn the common terminology of interest groups and research communities. In: Proc. 9th Conf on Terminology and Artificial Intelligence, TIA 2007, Sophia Antinopolis, France (2007)
15. Kozakov, L., Park, Y., Fin, T., Drissi, Y., Doganata, Y., Cofino, T.: Glossary extraction and utilization in the information search and delivery system for IBM Technical Support. IBM System Journal 43(3), 546--563 (2004). doi: 10.1147/sj.433.0546
16. Astrakhantsev, N.: Methods and software for terminology extraction from domain-specific text collection. PhD thesis, Institute for System Programming of Russian Academy of Sciences (2015)

17. Bordea, G., Buitelaar, P., Polajnar, T.: Domain-independent term extraction through domain modelling. In: Proc. 10th Int Conf on Terminology and Artificial Intelligence, TIA 2013, Paris, France (2013)
18. Park, Y., Byrd, R. J., Boguraev, B.: Automatic glossary extraction: beyond terminology identification. In: Proc. 19th Int Conf on Computational linguistics, pp. 1--7. Taipei, Taiwan (2002) doi: 10.3115/1072228.1072370
19. Nokel, M., Loukachevitch, N.: An experimental study of term extraction for real information-retrieval thesauri. In: Proc 10th Int Conf on Terminology and Artificial Intelligence, pp. 69--76 (2013)
20. Zhang, Z., Gao, J., Ciravegna, F.: Jate 2.0: Java automatic term extraction with Apache Solr. In: Proc.LREC 2016, pp. 2262--2269, Slovenia (2016)
21. Justeson, J., Katz, S. M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1(1), 9--27 (1995). doi: 10.1017/S1351324900000048
22. Evans, D. A., Lefferts, R. G.: Clarit-trec experiments. Information processing & management 31(3), 385--395 (1995). doi: 10.1016/0306-4573(94)00054-7
23. Church, K. W., Gale, W. A.: Inverse document frequency (idf): a measure of deviations from Poisson. In: Proc. ACL 3rd Workshop on Very Large Corpora, pp. 121--130, Association for Computational Linguistics, Stroudsburg, PA, USA (1995) doi: 10.1007/978-94-017-2390-9_18
24. Oliver, A., V`azquez, M.: TBXTools: a Free, Fast and Flexible Tool for Automatic Terminology Extraction. In: Angelova, G/, Bontcheva, K., Mitkov, R. (eds.): Proc. Recent Advances in Natural Language Processing, pp. 473-479, Hissar, Bulgaria, Sep. 7-9 (2015)
25. Corcho, O., Gonzalez, R., Badenes, C., and Dong, F.: Repository of indexed ROs. Deliverable No. 5.4. Dr Inventor project (2015)
26. Ermolayev, V., Batsakis, S., Keberle, N., Tatarintseva, O., Antoniou, G.: Ontologies of Time: Review and Trends. International Journal of Computer Science and Applications 11(3), 57--115 (2014)
27. Osborne, F., Salatino, A., Birukou, A., Motta, E.: Automatic Classification of Springer Nature Proceedings with Smart Topic Miner. In: Groth, P. et al. (Eds.) ISWC 2016, LNCS, vol. 9982, pp. 383--399 (2016) doi: 10.1007/978-3-319-46547-0_33