Cross-Evaluation of Automated Term Extraction Tools by Measuring Terminological Saturation

Victoria Kosa[0000-0002-7300-8818, 1], David Chaves-Fraga[0000-0003-3236-2789, 2],
Dmitriy Naumenko[0000-0001-5678-4439, 3], Eugene Yuschenko[0000-0003-4523-6933, 3],
Carlos Badenes-Olmedo[0000-0002-2753-9917, 2], Vadim Ermolayev[0000-0002-5159-254X, 1],
and Aliaksandr Birukou[0000-0002-4925-9131, 4]

[1] Department of Computer Science, Zaporizhzhia National University,
Zhukovskogo st. 66, Zaporizhzhia, Ukraine
victoriya1402.kosa@gmail.com, vadim@ermolayev.com
[2] Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
{dchaves, cbadenes}@fi.upm.es
[3] BWT Group, Mayakovskogo st. 11, Zaporizhzhia, Ukraine
admin@groupbwt.com
[4] Springer-Verlag GmbH, Tiergartenstrasse 17, Heidelberg, Germany
aliaksandr.birukou@springer.com

Abstract. This paper reports on cross-evaluating the two software tools for automated term extraction (ATE) from English texts: NaCTeM TerMine and UPM Term Extractor. The objective 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 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 paper presents the suite of instrumental software modules, experimental workflow, 2 synthetic and 3 real document collections, generated datasets, and set-up of our experiments. Next, the results of the cross-evaluation experiments are presented, analyzed, and discussed. Finally the paper 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, Retrospective Document Collection, OntoElect

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 significant 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 paper[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.

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 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 6 present our contributions.

---

[1] This paper is based on [1] in terms of its idea and research agenda presented as its research hypothesis and questions in Section 2. The rest constitutes the new result elaborated after the submission and publication of [1].

We focused our experiments on a single but important factor that may influence terminological saturation – the choice of an ATE software tool. Further, we presented our generic workflow to support different series of experiments answering different research questions in our project [1]. We also developed the suite of instrumental software modules to support our experimental workflow. 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.

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 5 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 6.

Finally, we summarize our results in Section 7, which concludes the paper.

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.

In our project we put forward a hypothesis that terminological saturation in a collection of documents is a complex thing which may depend on several aspects. 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 these documents indeed provide the biggest terminological contribution to the document collection?

---

[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.

- **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. Thus, the setting of the experiments should consider these aspects.

In this paper we aim at finding out the answer to our research question **Q1**: which relevant term extraction software yields the best (smallest) saturated sub-sets of documents? Therefore, the rest of the paper is focused around this aspect.

We review the related work along the following lines. We compare 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 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], 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 some of them that 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 of 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].

A 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 that 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, nc-value	+	-	0.73; **0.53**	1.00
						0.77; **0.56**	1.00
Weird-ness [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

Method [Source]	Domain-indepen-dence (+/-)	Super-vizion (U/SS)	Metrics	Term Signi-ficance	Cut-off (+/-)	Precision (GENIA; average)	Run Time (%/c-value)
TermEx [14]	+	SS	Domain Perti-nence, Domain Consensus, Lexi-cal Cohesion, Structural Rele-vance	-	+	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 par-ticular domain. A domain-independent method is sought as our aim is to develop a domain-independent technique.

Supervision: "U" – unsupervised; "SS" – semi-supervised. An unsupervised method is sought as our aim is to develop an unsupervised technique.

Term Significance: "+" – the method returns a value for each retained term which could fur-ther 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.

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

For us, only the methods are relevant that do not do cut-offs and return significance values. Our THD algorithm does cut-offs at a later stage.

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 Run Time column.

After analyzing the findings listed in Table 1, we support the conclusion of [20] stating that "*c-value* is the most reliable method as it obtains consistently good results, in terms of precision", evenly on the two different mixes of datasets – [20] and [4]. We also note that *c-value* is one of the slowest in the group of unsupervised and do-main-independent methods, though its performance is comparable with the fastest ones. Still, *c-value* outperforms the domain-specific methods, sometimes signifi-cantly, as it is in the case with PU-ATR. Hence, we have chosen *c-value* as the

method for our cross-evaluation experiments. We were therefore looking further at the tools which implemented *c-value* and were 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 descriptions of term extraction tools at several 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 outlined 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		Biomedical	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 (TaaS EU Project)	https://term.tilde.com/	Identifies term candidates in documents and extracts 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, multi-lingual, requires language corpus	Deletes n-grams with stop words
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

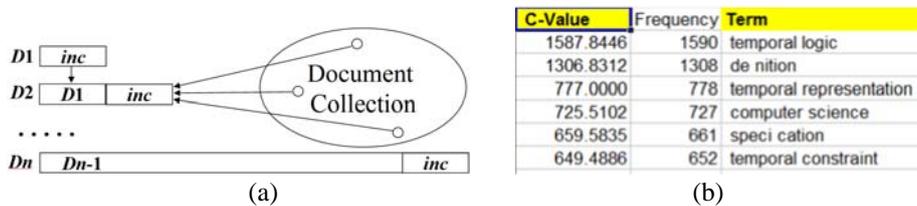For the final selection of the tools for our cross-evaluation we:
- Decided not to consider ATR4S and JATE 2.0, at list at this stage, because it was not fully clear how to extract the *c-value* method implementation from these suites
- Selected the tools that use the *c-value* method – which are NaCTeM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

3 OntoElect Saturation Metric and Measurement Pipeline

OntoElect methodology [2] seeks for maximizing the fitness of the developed ontology to what the domain knowledge stakeholders think about the domain. Fitness is measured as the stakeholders' "votes" which allows assessing 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. All the constituents of OntoElect as a processing technique are formally presented in [2].

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 OntoElect, 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 a 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 a document collection by analyzing the *saturation* of terminological footprints of the incremental slices of the collection, as e.g. reported in [26]. The full texts of the documents from the retrospective collection are grouped in datasets in the increasing order of their timestamps. As pictured in Fig. 1a, the first dataset $D1$ contains the first portion (*inc*) of documents. The second dataset $D2$ contains the first dataset $D1$ 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	de nition
777.0000	778	temporal representation
725.5102	727	computer science
659.5835	661	speci cation
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 terms $B1$, $B2$, …, Bn are extracted from the datasets $D1$, $D2$, …, Dn, using TerMine software, together with
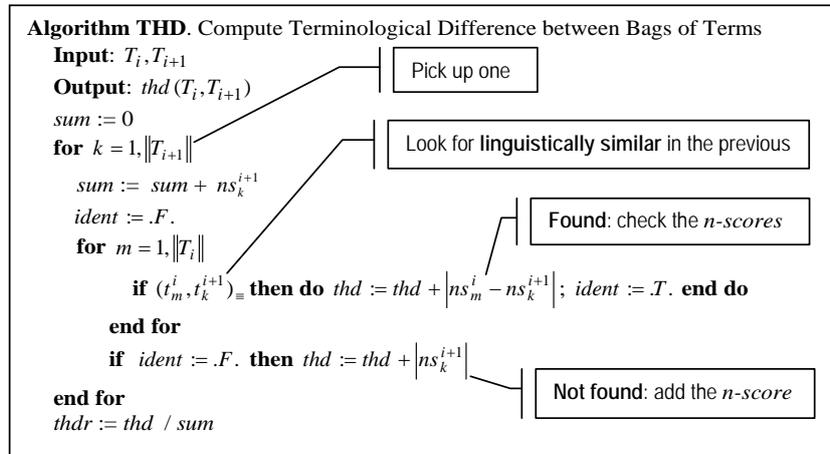
their *significance* (*c-value*) scores. Please see an example of a bag of terms extracted by TerMine in Fig. 1b.

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 retain those terms that are 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* – the bags of retained terms

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 reader convenience.

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.

Algorithm THD. Compute Terminological Difference between Bags of Terms
Input: T_i, T_{i+1}
Output: $thd(T_i, T_{i+1})$
$sum := 0$
for $k = 1, \|T_{i+1}\|$
  $sum := sum + ns_k^{i+1}$
  $ident := .F.$
  for $m = 1, \|T_i\|$
    if $(t_m^i, t_k^{i+1})_{\equiv}$ **then do** $thd := thd + \left|ns_m^i - ns_k^{i+1}\right|$; $ident := .T.$ **end do**
  end for
  if $ident := .F.$ **then** $thd := thd + \left|ns_k^{i+1}\right|$
end for
$thdr := thd / sum$

Pick up one

Look for **linguistically similar** in the previous

Found: check the *n-scores*

Not found: add the *n-score*

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

Absolute (*thd*) and relative (*thdr*) terminological differences are computed for further assessing if *Ti*+1 differs from *Ti* more than by 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.

To finalize this concise presentation of the OntoElect approach, it is worth noting that it is domain independent and unsupervised – due to the use of TerMine for term extraction. The shortcomings of this reliance on TerMine are revealed in our experimental study (Section 6).

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 Experimental Workflow and Software Tools

In this section we present our generic experimental workflow and the suite of instrumental software tools which have been developed to support our experiments.

4.1 Generic Experimental Workflow and Instrumental Software

Our generic experimental workflow, outlined in Fig. 3, is based on the OntoElect processing pipeline (Section 3). In particular, this workflow will be applied (using Configure Experiment step) to perform all the cross-evaluation experiments described below (Section 6).

The workflow covers the preparatory phase, experiment configuration, the generation of the datasets, term extraction, saturation measurement, and the analysis and comparison 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. To support the rest of the steps, the instrumental software has been developed and offered for public use – as described in [28].

The **preparatory** phase includes:
- The **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 Scholar[4]. This step is supported by the **Catalogue Generator** module.
- The **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. This step is supported by the **Full Text Downloader** module.
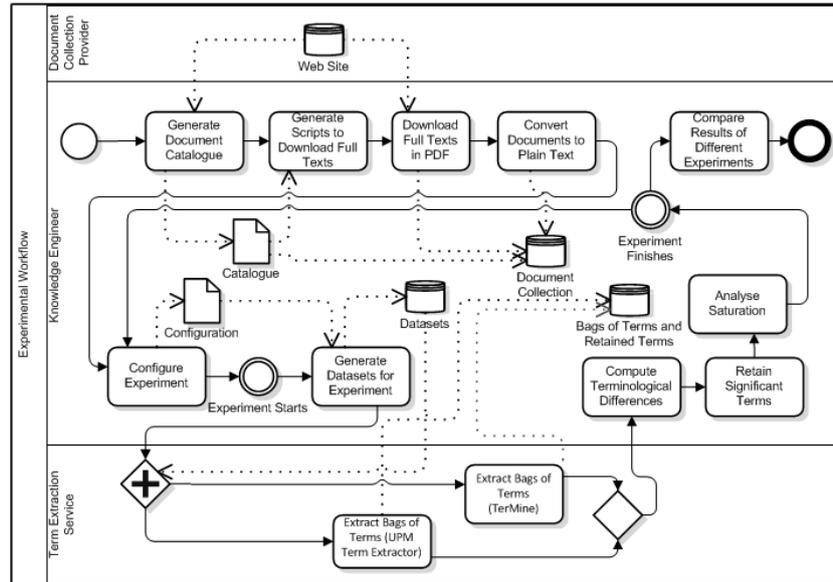

Fig. 3: Experimental workflow

- The **conversion** of the full texts of the downloaded documents to the **plain text** format for further term extraction is supported by the **PDF to Plain Text Convertor** module

  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 (Section 3), to be further taken by term extraction. The texts are added to the increments in the order chosen as the parameter of the experiment. This phase is supported by the **Dataset Generator** module.

  The phase of **term extraction** applies the chosen software tool to the generated datasets: $D1$, $D2$, … In result, it outputs the bags of extracted terms $B1$, $B2$, … In the context of reported experiments, this phase is supported by the use of the two software tools: **UPM Term Extractor** and **NaCTeM TerMine**. UPM Term Extractor has been developed in the Dr Inventor project. The tool takes a collection of documents (PDF or plain text) in English or a dataset generated from this collection (plain text) 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[5]. It takes an English plain text (ANSI) document (dataset) 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* (TTF).

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 (see [28] for more details). This phase is supported by the **THD** modules, the **Convertor** module, and **StopTermRemover** module. The **THD modules** implement the THD algorithm for the input bags of terms in UPM Term Extractor and NaCTeM TerMine formats. The **Convertor** takes a bag of terms in TerMine format and saves it in the UPM Extractor format. The **StopTermRemover** takes the list of the manually selected stop terms and deletes all these terms from the bags of terms.

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

Hence, our experimental workflow is fully covered by the developed and used instrumental software.

4.2 Planned Series of Experiments

Different series of experiments, using this workflow, are planned to be conducted in the presented project [1].

The **first series** are planned for experimental cross-evaluation of the selected ATE 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 our research question **Q1** (c.f. Section 2).

For this we set-up the first series of experiments to cross-evaluate UPM Term Extractor versus NaCTeM TerMine. In this subsection we present the configuration of these experimental series and the measurements in more detail.

We plan to perform this cross-evaluation by applying the experimental workflow to the 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. All the document collections are presented in more detail in Section 5.

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 the similar bags of terms from each of the document collections in which saturation has been observed. The similarity between the extracted bags

---

of terms is also measured using *thd* versus *eps* approach by applying the THD module to the pairs ($B1$, $B1m$), ($B2$, $B2m$), …, (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).

5 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[6].

5.1 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 at these two extremes. If so, there is more confidence that it is also adequate for real document collections.

1DOC is the document collection containing just one paper. As this paper, we used the source of [24]. It has been converted to plain ANSI text format manually. From the plain text, the datasets $D1$, $D2$, …, $D20$ 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 quite quickly with *thd* close to 0, as all the increments are identical.

The intuition behind crafting the RAW collection is opposite to the previous case. To avoid saturation, a collection is required in which all the increments are substantially terminologically different. To have that, the documents dealing with different topics, coming from different fields, and therefore using very different terminology need to be put together. For constructing RAW 80 articles from English Wikipedia have been randomly selected 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, these PDF files have been converted to plain ASCII texts using our PDF to Plain Text Convertor. 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, 20 datasets have been generated, $D1$, $D2$, …, $D20$, with increments comprising 4 randomly taken documents from the collection.

5.2 Real Document Collections

Our real document collections are all composed of the 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[7]; the DMKD collection is composed of the subset of full text articles from the Springer journal on Data Mining and Knowledge Discovery[8]; the DAC collection comprises the subset of full text papers of the Design Automation Conference[9].

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 [24]. It contains all the papers published in the TIME symposia proceedings between 1994 and 2013, which are 437 full text documents. These papers 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 $D1, D2, …, D22$.

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. It was provided by Springer based on their policy on full text provision for data mining purposes[10]. To the DMKD document collection, we have included 300 papers published in the Journal of Data Mining and Knowledge Discovery between 1997 and 2010. All the papers in their full texts were automatically processed using our instrumental pipeline. 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 20 papers. So, based on the available documents we have generated 15 incrementally enlarged datasets $D1, D2, …, D15$.

The domain of the DAC collection is Engineering Design Automation. The publisher of these papers is IEEE. For this collection, we have chosen 506 papers published between 2004 and 2010. The papers of DAC 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 $D1, D2, …, D26$.

---

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

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

[9] http://dac.com/

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

5.3 Summary of Data Features

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

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

Collection	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	20 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.

6 Experiments and Discussion

In this section we report and discuss the results of our experiments on the datasets generated from the five data collections presented in Section 5, particularly on the results of the phases of term extraction, saturation measurement, analysis and comparison.

In the experiment with each collection we: (i) extracted the bags of terms from the prepared datasets using TerMine and UPM Extractor; (ii) measured saturation for both sets of the bags of terms using the corresponding THD modules; (iii) measured comparative saturation for the pairs of the bags of terms ($B1$, $B1m$), ($B2$, $B2m$), …, (Bn, Bnm) – as described in Section 4.2; (iv) 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 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 Re-

mover module. So, for the RAW collection we also compared noisy and de-noised bags of terms.

6.1 Terminological Saturation in Synthetic Collections

Due to collections design (Section 5), the results on 1DOC are expected to demonstrate quick and steady saturation and the results on RAW have to be far from being saturated.

For the bags of terms extracted from 1DOC, 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 visualized in Fig. 4(a)[11]. We then measured 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 pictured in Fig. 4(b).
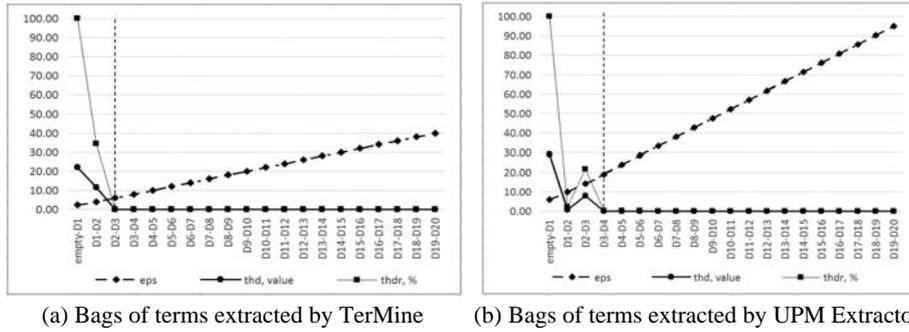

(a) Bags of terms extracted by TerMine   (b) Bags of terms extracted by UPM Extractor

Fig. 4: Visualization of saturation measurements on the 1DOC datasets

The dashed vertical line in Fig. 4(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 steady 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*2-*T*3. The dashed vertical line in Fig. 4(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.

---

[11] The values measured in all the reported experiments, though sometimes mentioned in the text, are not presented in the paper for saving space. All these experimental data and results are presented in full detail in the supporting technical report [28] which is publicly available online.

The differences in saturation measurements for the bags of terms extracted by Ter-Mine and UPM Extractor 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) ididual term significance thresholds (*eps*) were about 2.5 times higher for UPM Extractor; (iii) the number of retained terms with *c-value* > *eps* was ~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, compared to UPM Extractor results. 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 ~3 times more selective regarding extracting term candidates. So, the pre-processing in Ter-Mine is more sophisticated. 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 selective instrument.
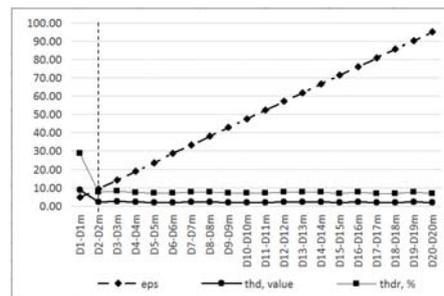
Fig. 5: Comparison of the retained sets of terms extracted from the 1DOC collection by UPM Extractor and TerMine

It was further checked if both tools extracted statistically similar sets of terms from the 1DOC collection. The measurements are visualized in Fig. 5. The figure shows 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 RAW 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 visualized in Fig. 6(a).

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 ~200 stop terms. These stop terms have been removed from the bags of terms *B*1, …, *B*20 and saturation analysis has been repeated. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) for so de-noised bags of terms are visualized in Fig. 6(b).
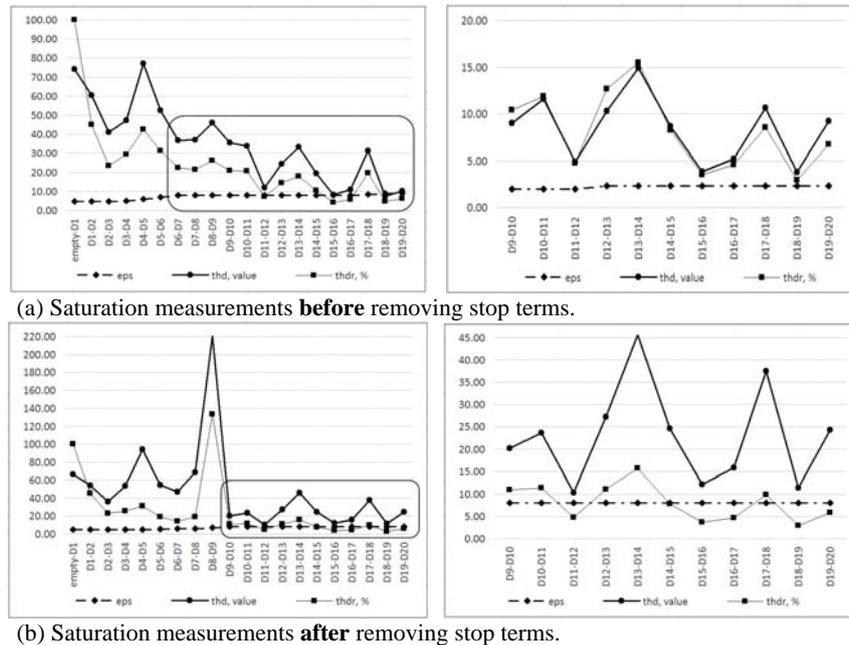
When looking at Fig. 6(a) and, especially, at 6(b), we observe that, as it was expected, the RAW collection is not terminologically saturated. Further, looking at the differences between Fig. 6 (a) and (b), we observe some nice indicators of the presence of noise in the textual documents of the collection. Indeed, the *thdr* values in Fig. 6(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 similar to that of *thdr*. At the same time the *thd* and *thdr* curves in Fig 6(b) very much resemble the *thdr* curve in Fig. 6(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.

(a) Saturation measurements **before** removing stop terms

(b) Saturation measurements **after** removing stop terms
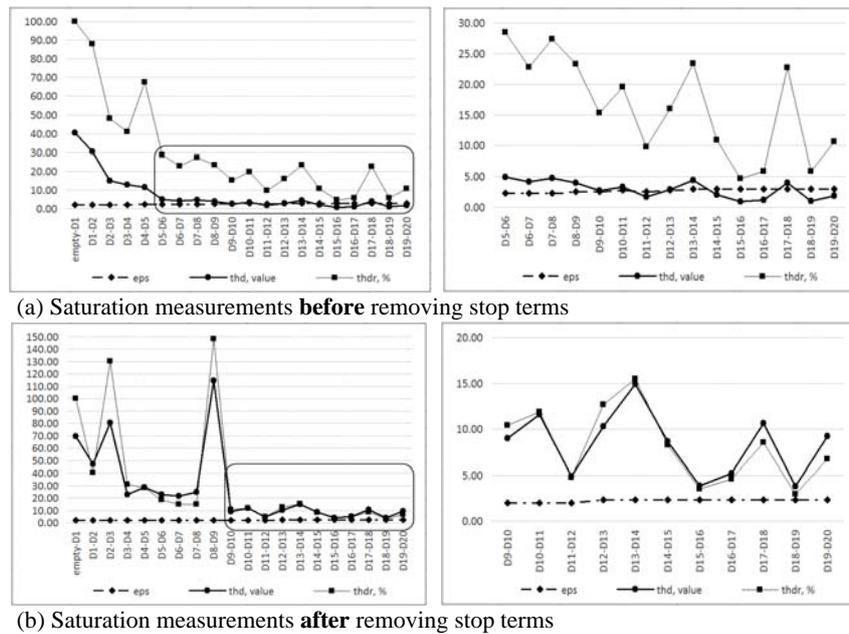
Fig. 6: Visualization of saturation measurements on the RAW bags of terms extracted by TerMine. The diagram to the right represents a more granular look into the rounded rectangle in the diagram to the left.

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

The values of individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) are visualized in Fig. 7(a). 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 ~220 stop terms. These stop terms have been removed from the bags of terms *B*1, …, *B*20 and saturation analysis has been repeated. The values of individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) for so de-noised bags of terms are pictured in Fig. 7(b).

(a) Saturation measurements **before** removing stop terms.

(b) Saturation measurements **after** removing stop terms.

Fig. 7: Visualization of saturation measurements on the RAW bags of terms extracted by UPM Extractor.

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 clearer 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, the use of UPM Extractor is preferred in this case as it is a sharper instrument.

For this collection, it has not been measured if both tools extract statistically similar bags of terms. This measurement would have had no value in the absence of saturation.

6.2 Terminological Saturation in Real Collections

Our results in measuring terminological saturation in the real document collections are presented and analyzed in this subsection.

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

The bags of terms extracted by TerMine were first processed. The results of measuring individual term significance thresholds (*eps*) and terminological differences (*thd*, *thdr*) are visualized in Fig. 8. The diagram at the left visualizes the whole set of measures. 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*.
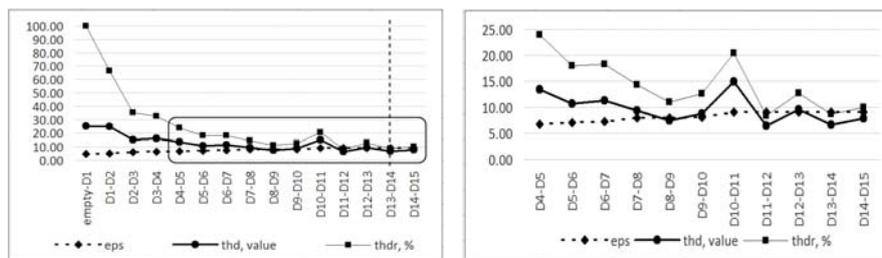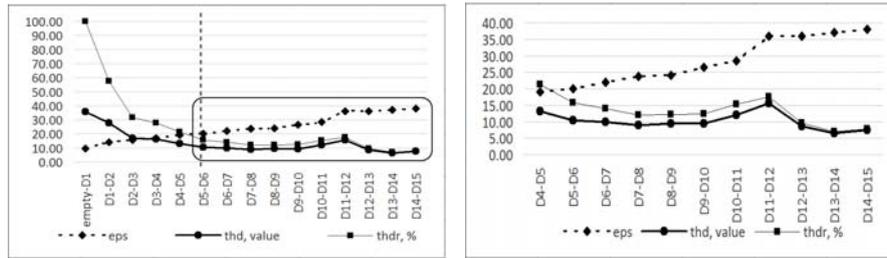

Fig. 8: Saturation measurements on the DMKD datasets based on the bags of terms extracted by TerMine

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 compared to the vicinity – please see DAC results for much higher peaks. Overall, it is too early to consider DMKD saturated based on the extraction results by TerMine.

The results of measuring saturation based on the bags of terms extracted by UPM Extractor are pictured in Fig. 9. It could be noted that steady 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 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

Extractor results. Saturation is reached due to much higher values of individual term significance threshold *eps*.

Hence, for this document collection, **UPM Extractor** yields **better circumscribed** and **more compact** sets of **significant terms** and the cut-off happens at much higher values of term significance.

Fig. 9: Saturation measurements on the DMKD datasets based on the bags of terms extracted by UPM Extractor

One hypothesis about the reason for better UPM Extractor performance could be that it extracts not all the terms from the documents it takes in, and TerMine reaches a substantially higher recall. 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 pictured diagrammatically in Fig. 10.
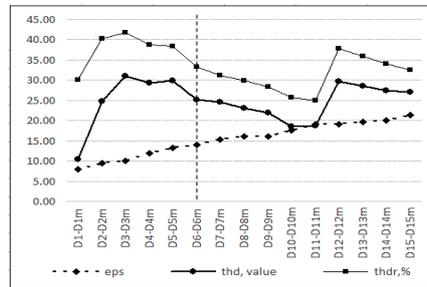
Fig. 10: Comparison of the retained sets of terms extracted from the DMKD collection by UPM Extractor and TerMine

Fig. 10 shows that both tools extract somewhat similar bags of terms. This similarity increases with the growth of a dataset. The numbers of retained terms are higher than in Fig. 8 and 9. These also hint that the extracted bags of terms are similar and recall values of individual tools differ not too much, which is acceptable.

Interestingly, terminological difference (*thd*) in Fig. 10 goes below *eps* exactly at the point when TerMine results show the highest terminological peak (c.f. Fig. 8). 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[12] that the reason for that is the increasing influence of the accumulated noise in the datasets, which is perceived differently by the individual tools.

The results of saturation measurements for TIME are pictured in Fig. 11.

The saturation measurements based on the bags of terms extracted by TerMine **did not show any saturation** –Fig. 11(a). The *thd* values did not go below *eps*. The tendency is similar to the DMKD experiment – 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 vicinity.
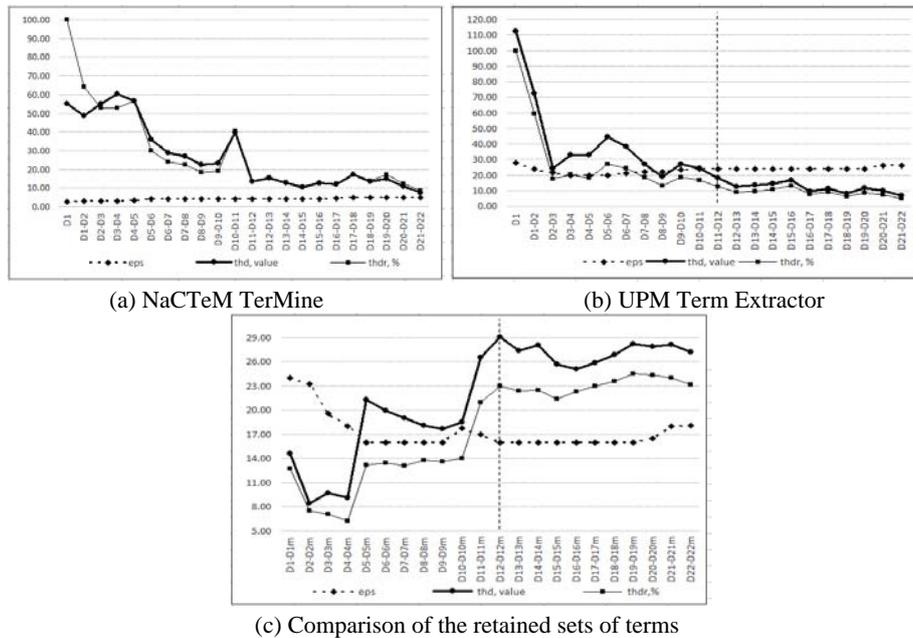

(a) NaCTeM TerMine   (b) UPM Term Extractor

(c) Comparison of the retained sets of terms

Fig. 11: Saturation measurements for the TIME collection

The saturation measurements based on the bags of terms extracted by UPM Extractor **reveal stable saturation** starting from *D*11-*D*12 – as pictured in Fig. 11(b) by the vertical dashed line. The values of *thd* and *thdr* resemble these of the TerMine case,

---

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

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 $B12$ is 7110, which is only 2.47% of the total number of extracted terms in $B12$. 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 TIME, **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.

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 and are pictured in Fig. 11(c). It could be seen, that the terminological difference (*thd*) between the bags of retained terms at the saturation point $D12$-$D12m$[13] 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 DAC are shown in Fig. 12. DAC collection is much noisier than DMKD and TIME. The results also differ – in values but not in the overall picture.

The saturation measurements based on the bags of terms extracted by TerMine revealed the potential saturation point only in the last measurement at $D25$-$D26$ – as pictured in Fig. 12(a). However, the terminological peak at $D24$-$D25$, 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. Fig. 11(a) and 12(a), if compared, show the differences in measurement values for the dataset pairs of roughly similar sizes.

The comparison of the measurements for TIME and DAC, based on the extraction results by TerMine, reveals that: (i) the values of *eps* grow faster for TIME than for DAC; (ii) the numbers of extracted and retained terms for DAC are substantially higher than for TIME; (iii) the numbers of retained terms for TIME grow monotonically and this growth slows down – an indicator of possible saturation in the upcom-

---

[13] $D12$ is the dataset from which $B12$ is extracted by UPM Extractor and $B12m$ by TerMine. $B12m$ is further converted to the UPM Extractor format and the pair ($B12$, $B12m$) is fed into the THD module. The module returns *eps*, *thd*, and *thdr* values for the pair as described in Section 3.

ing measurements; (iv) the number of retained terms for DAC substantially drops below the previous value at *D*24-*D*25 and the *thd* dramatically picks up from 21.51 to 135.49. We believe, again, that the reason for the peak at *D*24-*D*25 is the influence of the accumulated noise. However, TerMine signals about the problem quite lately.

(a) NaCTeM TerMine   (b) UPM Term Extractor

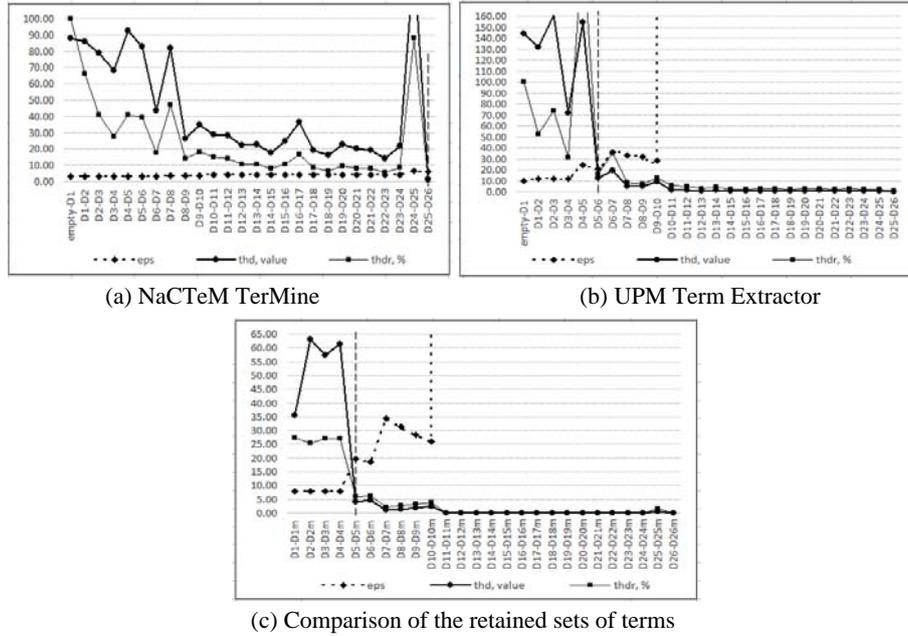(c) Comparison of the retained sets of terms

Fig. 12: Saturation measurements for the DAC collection

The saturation measurements based on the bags of terms extracted by UPM Extractor **reveal steady saturation** starting from *D*5-*D*6 with *eps* at about 20 – as pictured in Fig. 12(b) by the vertical dashed line. However, 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 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 Fig. 12(b) could be used as an indicator of the need to de-noise the collection datasets before terminology extraction.

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 pictured in Fig. 12(c). 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.

6.3 Summary of Results and Recommendations

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

Conclusions (case 1): (i) linguistically, TerMine is more selective in extracting term candidates, (ii) the cut-offs in UPM Extractor outputs happen for substantially more significant terms; (iii) UPM Extractor circumscribes more compact, yet more significant sets of terms and is a more sensitive instrument; (iv) these results confirm the adequacy of our saturation metric for the boundary case of quick saturation.

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.

Conclusions (case 2): (i) TerMine is more sensitive in indicating the need to denoise the bags of terms; (ii) UPM Extractor is more sensitive in detecting the absence of saturation; (iii) these results confirm the adequacy of our saturation metric for the boundary case of non-reachable saturation

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

Case 3: DMKD (automatically pre-processed). Overall, it cannot be reliably judged that the DMKD collection is saturated based on the extraction results by Ter-

Mine. In difference to that, the saturation measurements using the bags of terms extracted by UPM Extractor reveal steady saturation quite quickly. It has also been noticed that both tools extracted statistically similar bags of terms.

Case 4: TIME (manually de-noised). Saturation measurements using the bags of terms extracted by TerMine failed to detect saturation in the TIME collection. Very similarly to the DMKD case, the saturation measurements using the bags of terms extracted by UPM Extractor reveal steady saturation quite quickly, also with much higher individual term importance thresholds *eps*. These result in significantly more compact sets of retained significant terms.

Conclusion (cases 3, 4): Both cases demonstrated similar advantages of UPM Extractor over TerMine in detecting saturation and retaining significant terms. In both cases UPM Term Extractor yielded better circumscribed and more compact sets of significant terms. Manual cleaning of the TIME collection did not help noticeably to improve the results of saturation measurements, therefore was not really necessary.

Case 5: DAC (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, 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 (cases 3-5): 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.

7 Conclusions and Future Work

This paper reported on cross-evaluating the two software ATE tools: NaCTeM TerMine and UPM Term Extractor. The tools were selected for cross-evaluation based on the analysis of the related work in ATE and availability of software as reported in Section 2.

The objective of our cross-evaluation experiments 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 text document collections in an arbitrary domain of interest. The technique for measuring terminological saturation, based on the use of the THD algorithm, has been outlined in Section 3.

The paper presented the set-up of experiments by outlining the generic workflow and instrumental software tools developed to automate the activities in the workflow, such as document collection retrieval, pre-processing, dataset generation, term extraction, terminological difference measurement, bags of terms de-noising. It also explained which kinds of measurements and observations were planned to cross-

evaluate the fitness of the selected ATE tools for their use in terminological saturation measurement pipeline. Specifically we were interested in: (i) how quickly the bags of terms, extracted, by different tools, from the incrementally growing datasets, saturated terminologically in terms of *thd* versus *eps*; and (ii) if the tools extracted the similar bags of terms from the document collections.

The paper then presented the data collections which were used in the experiments. The experiments were first been planned on the two synthetic collections to find out if the measurements of terminological saturation are adequate in the boundary cases: (i) the 1DOC collection in which saturation should be detected swiftly; and (ii) the RAW collection in which terminology can not be saturated. Secondly, the experiments were planned on the three real document collections, DMKD, TIME, and DAC. These collections represent different domains. The documents in these collections had different layouts and were processed differently, leaving more or less noise in the datasets. The summary of the collection and dataset features was provided in Table 3.

Finally, the results of our experiments on the datasets generated from the five data collections, particularly on the results of the phases of term extraction, saturation measurement, analysis and comparison, were reported and discussed. Based on the analysis of experimental results, conclusions were made in subsection 6.3. The conclusions revealed that:

- The metrics we used to measure terminological saturation are adequate as the results in the boundary cases of 1DOC and RAW were as expected
- The use of UPM Extractor is preferred, over TerMine, to detect that saturation is hardly expected, like in the RAW case
- When terminological saturation was reachable, the bags of terms extracted by UPM Extractor converged to saturation quicker than that by TerMine. Their use yielded better circumscribed and more compact sets of significant terms.
- In the cases of noisy datasets and due to not being very selective in extracting term candidates, UPM Extractor was much more sensitive in detecting excessive noise, compared to TerMine.

Based on these conclusions it has been recommended that the use of UPM Term Extractor is more preferable than the use of NaCTeM Termine in our terminological saturation measurement and detection technique.

Our future work will follow the research agenda outlined in Section 2 as the list of research questions **Q2**-**Q5**. Currently[14], the series of experiments aimed at answering **Q2** are finished and the technical report is being written. Our next step will be to configure and perform the experiments for answering **Q3**. The answers to **Q4** and **Q5** are in our mid-term plans for the future work.

Acknowledgements

---

[14] At the time of writing the final version of this paper, December, 2017.

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

28. Kosa, V., Chaves Fraga, D., Naumenko, D., Yuschenko, E., Badenes, 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. (2017) online: http://ermolayev.com/TS-RTDS-TR-2017-1.pdf DOI: 10.13140/RG.2.2.31187.07207