Review

Review of Tools for Semantics Extraction: Application in Tsunami Research Domain

František Babič [1], Vladimír Bureš [2,*], Pavel Čech [2], Martina Husáková [2], Peter Mikulecký [2], Karel Mls [2], Tomáš Nacházel [2], Daniela Ponce [2], Kamila Štekerová [2], Ioanna Triantafyllou [3], Petr Tučník [2] and Marek Zanker [2]

[1] Faculty of Electrical Engineering and Informatics, Technical University of Košice, 042 00 Košice, Slovakia; frantisek.babic@tuke.sk
[2] Faculty of Informatics and Management, University of Hradec Králové, 500 03 Hradec Králové, Czech Republic; pavel.cech@uhk.cz (P.Č.); martina.husakova.2@uhk.cz (M.H.); peter.mikulecky@uhk.cz (P.M.); karel.mls@uhk.cz (K.M.); tomas.nachazel@uhk.cz (T.N.); daniela.ponce@uhk.cz (D.P.); kamila.stekerova@uhk.cz (K.Š.); petr.tucnik@uhk.cz (P.T.); marek.zanker@uhk.cz (M.Z.)
[3] Department of Geology and Geoenvironment, National & Kapodistrian University of Athens, 106 79 Athens, Greece; ioannatriantafyllou@yahoo.gr
* Correspondence: vladimir.bures@uhk.cz; Tel.: +420-493332259

Abstract: Immense numbers of textual documents are available in a digital form. Research activities are focused on methods of how to speed up their processing to avoid information overloading or to provide formal structures for the problem solving or decision making of intelligent agents. Ontology learning is one of the directions which contributes to all of these activities. The main aim of the ontology learning is to semi-automatically, or fully automatically, extract ontologies—formal structures able to express information or knowledge. The primary motivation behind this paper is to facilitate the processing of a large collection of papers focused on disaster management, especially on tsunami research, using the ontology learning. Various tools of ontology learning are mentioned in the literature at present. The main aim of the paper is to uncover these tools, i.e., to find out which of these tools can be practically used for ontology learning in the tsunami application domain. Specific criteria are predefined for their evaluation, with respect to the "Ontology learning layer cake", which introduces the fundamental phases of ontology learning. ScienceDirect and Web of Science scientific databases are explored, and various solutions for semantics extraction are manually "mined" from the journal articles. ProgrammableWeb site is used for exploration of the tools, frameworks, or APIs applied for the same purpose. Statistics answer the question of which tools are mostly mentioned in these journal articles and on the website. These tools are then investigated more thoroughly, and conclusions about their usage are made with respect to the tsunami domain, for which the tools are tested. Results are not satisfactory because only a limited number of tools can be practically used for ontology learning at present.

Keywords: semantics; taxonomy; ontology; ontology learning; knowledge extraction; natural language processing; tool; tsunami

check for updates

Citation: Babič, F.; Bureš, V.; Čech, P.; Husáková, M.; Mikulecký, P.; Mls, K.; Nacházel, T.; Ponce, D.; Štekerová, K.; Triantafyllou, I.; et al. Review of Tools for Semantics Extraction: Application in Tsunami Research Domain. *Information* **2022**, *13*, 4. https://doi.org/10.3390/info13010004

Academic Editors: Riccardo Albertoni and Peter Winstanley

Received: 12 November 2021
Accepted: 20 December 2021
Published: 24 December 2021

1. Introduction

A large volume of unstructured texts occurs daily, especially in the web space. Valuable information and knowledge are spread across this extensive collection of texts. The big challenge is to "mine" helpful information and knowledge for problem solving or decision making without human effort. Sophisticated and efficient methods of auto-mated text processing can deal with this problem. In a general point of view, techniques of linguistics, statistics, and artificial intelligence are used for semi-automatic or fully automatic extraction of information and knowledge from structured (spreadsheets-like), semi-structured (markup-like, JSON-like), or unstructured (plain texts, PDF or Word files) collections of data. These techniques are cited in connection with the multidisciplinary research area called text mining. Text mining, as a subarea of the artificial intelligence, combines techniques and methods of data mining, machine learning, library and information sciences, (computational) linguistics, statistics, or databases for "turning texts into numbers" for handling texts, including manipulation with individual words, from sentences to whole documents [1]. Semantics plays a crucial role in the automated processing of texts. It is not only important to separate individual words or punctuation marks in the text (tokenization), in order to decide about the part of speech (parts of speech tagging) or to disambiguate words (morphological and lexical analysis), but it is also essential to detect relationships between these words in the context (semantic analysis). Taxonomy generation is a typical example where the main aim is to find more general and more specific concepts (words) from the text.

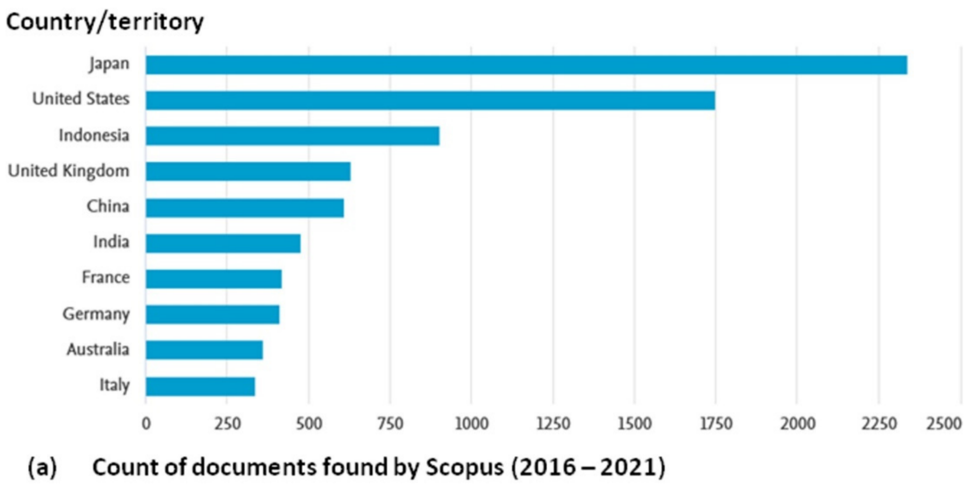
Different approaches and tools exist for (semi-)automated processing of texts [2]. If a user is familiar with programming, various computational libraries, frameworks, or application programming interfaces (APIs) can be used for detection and extraction semantics from texts. If a user does not have solid programming background or does not want "to waste time" by learning to program, tools having built-in text mining/natural language processing algorithms can be used. A user only loads a text or its collection (a corpus) into the repository of a tool, sets up the parameters, and runs the relevant procedures for text processing. Visual programming-based solutions lie in the middle of the above-mentioned approaches. The main aim of this paper is to provide a review of the tools mainly falling under the last category, i.e., to provide a review of tools assisting in automated text processing, where programming skills or a strong background of natural language processing or text mining-related concepts are not inevitable. The approach used in this review does not only require one to read scientific papers where tools have already been used, homepages of the tools, and additional materials but also requires one to test their functionalities. Reviews of text mining-related tools, including ontology learning or taxonomy induction, were already published in the past [3–7]. Yet these reviews are outdated, and most tools cannot currently be downloaded and used. If the review is up to date, the reader of the paper can find out that the tool cannot be downloaded, its installation is not trivial, or it is time-consuming. The main aim of the paper is to point out the current usability of these tools. Additional parameters investigated for tools are explained in detail in Sections 3 and 4 of the paper.

It is obvious that a large spectrum of text mining-related activities are used for information and knowledge extraction from texts. The aim of the paper is not to find the best tool for each of these text mining tasks, but it is to select a particular subset of tasks with respect to the needs of authors of the paper and to verify whether these tasks are provided by the tools. This subset of tasks (requirements) is presented in more details in the third and fourth sections of the paper. The tools are investigated with respect to the main interest of the authors—to automate the processing of a large collection of texts related to the tsunami research. This is the reason why a short piece of text focused on disaster management (tsunami event) is used to evaluate the tools. As a brief explanation, a tsunami is a series of long ocean waves which can be caused by earthquakes, volcanic eruptions, (submarine, onshore) landslides, or pressure disturbances in the air (a meteotsunami). Consequences of tsunami occurrence can be devastating. A mega-tsunami can damage coastlines and sweep vegetation with human habitations in a matter of seconds [8,9]. Not all tsunamis have such a hazardous impact. Some of them are only detected by sensitive instruments on the ocean floor (a micro-tsunami) [10]. It is clear that tsunami preparedness is crucial for survival.

Many scientific papers related to the tsunami can be found. As an example, the search in the Scopus scientific database with searching abstracts, keywords, and titles of research papers about "tsunami" in the time scale 2016–2021 yields 8834 documents (as of date of search: 14 September 2021). Scopus provides more detailed statistics about these documents, e.g., published documents per year (see Table 1), documents by country/territory, or documents by affiliation (see Figure 1). These brief statistics demonstrate that large collections of unstructured texts can be found in the tsunami research and make sense to automate their processing.

Table 1. Published documents in the Scopus per year (2016–2021).

Year	Count of Documents
2021	1072
2020	1568
2019	1574
2018	1551
2017	1453
2016	1616



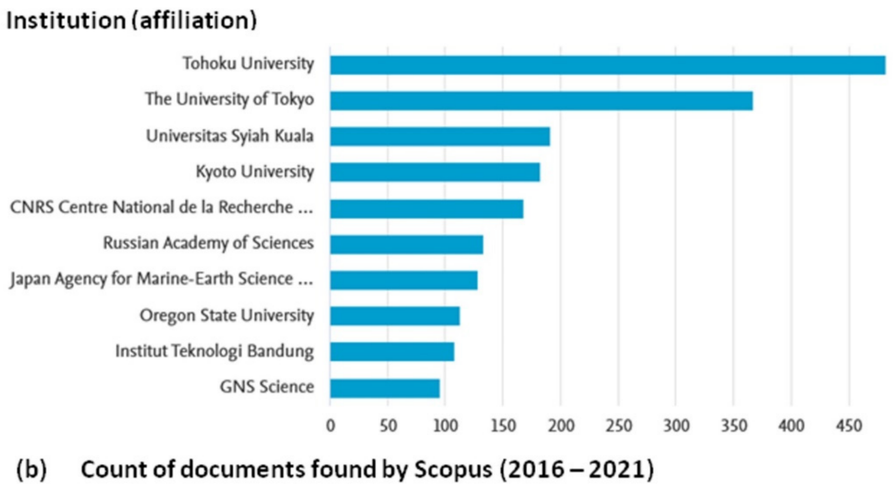


Figure 1. Visualisation of statistics about tsunami-related documents (Scopus search 2016–2021). (**a**) Found documents in the Scopus by country/territory. (**b**) Found documents in Scopus by an institution.

A researcher can read these papers one by one or can use a (semi-)automated tool as an assistant, e.g., for processing summary of the text, mining text having only positive opinion, identifying keywords and their inclusion into the taxonomy, clustering papers according to similarity, language detection, or identification the main topic(s) of the text. The main motivation behind this paper is to present a review of tools that could help the researchers to automatically extract interesting, valuable, and essential semantic structures related to the tsunami. The primary attention is paid to tools used for ontology learning.

The next aim of the paper is to provide a brief analysis of the selected ontology learning tools, where attention is paid to the ability to induce the taxonomy. The above-mentioned motivations and aims can be "condensed" into a "simple" research question: *"Which tools can be practically used for ontology learning at present (concerning the requirements mentioned in* Section 3.1*?)"*

The rest of the paper is organized as follows. Section 2 provides fundamental explanation of ontology learning. Section 3 presents a series of systematic steps which are realized for answering the question: "Which tools are going to be investigated for semantics extraction—ontology learning?". This section also presents criteria used for ontology learning tools evaluation. Section 4 presents analyses of research papers, where various tools for semantics extraction are mentioned. Results of evaluation of ontology learning tools are also presented in this section. Section 5 discusses the results and possible future research directions. Section 6 concludes the paper.

2. Semantics Extraction and Ontology Learning

The main idea behind the Semantic Web initiative is to develop standards and technologies assisting machines (intelligent agents) "to better understand" huge amounts of information available in the web space and autonomously perform sophisticated tasks on behalf of human users [11]. The key to success is to provide methods that could help to encode semantics which is behind the words. The architecture of the semantic web is based on traditional Web technologies and standards where:

(a) international characters sets are used for encoding information on the web (Uni-code)
(b) web sources are uniquely identified by the URI (Uniform Resource Identifier) and
(c) the XML (Extensible Markup Language) provides human-readable and machine-readable format for data exchange.

Ontologies are built on these layers and bring added value in structuring data on the web. The ontology is mainly perceived as a knowledge graph able to formally model different aspects of our real world. Intelligent agents use it as a vocabulary of "things". Thanks to its machine-processable structure, it facilitates communication between different parties existing in the web space. Formal ontology can be focused on the most fundamental concepts and relationships between them (an upper-level ontology), or its scope can be much more restricted on more specific concepts, which are part of the one application domain (a domain ontology).

Formal ontologies can be built completely manually, semi-automatically, or fully automatically [12]. Specialized ontological editors are mainly used in manual development of ontologies. Protégé editor [13] is de facto standard in ontological engineering assisting in manual development of ontologies and their management. In a general point of view, the manual approach of ontologies building is very time consuming. This is the main reason why there are tendencies to automate this process. Ontology learning (generation) is a multidisciplinary area consisting of techniques and methods of natural language processing, statistics, or linguistics, which are used for (semi-)automated acquisition of the formal ontologies using various sources. These sources (inputs) can be structured (database schemas, existing ontologies, knowledge bases), semi-structured (HTML or XML documents), or unstructured (a text expressed in natural language, e.g., Word documents, PDF documents) [4,14]. It is important to notice that two different aspects of ontologies development exist next to the ontologies learning: ontology population and ontology refinement. Both of these areas deal with already built ontological structure, which is less or more extended or updated. This paper is only focused on the ontology earning where the main goal is to build the new structure.

There are various approaches used for realization of the ontology learning. They can be categorized into the linguistic, statistical, or machine-learning approaches [3,15,16]. These ones lean against the so-called "Ontology learning layer cake" which visualises the phases of the ontology acquisition from unstructured texts [17,18]:

(a) extraction of the most important (representative) terms

(b) extraction of (multilingual) synonymous words from the previous step
(c) concepts extraction from the previous set of terms
(d) concept hierarchy extraction (extraction of IS-A (inheritance) relations)
(e) relationships extraction (extraction of non-taxonomic relations)
(f) relation hierarchy extraction
(g) extraction of axioms and rules (e.g., disjoint concepts)

Ontology learning is an immature research area. It is a challenge where the problem is to combine various techniques appropriately or to select the one which ensures the output corresponding to the reality or predefined requirements. This paper presents results of the review of papers from which the collection of tools used for ontology learning is extracted.

3. Materials and Methods

3.1. Process of Tools Selection

Figure 2 depicts how the final set of tools was obtained. Three sources are used for discovering tools for ontology learning: ScienceDirect scientific database (https://www.sciencedirect.com/, accessed on 10 October 2021), Web of Science scientific database (WOS) (https://www.webofscience.com/wos/woscc/basic-search, accessed on 10 October 2021), and ProgrammableWeb website (https://www.programmableweb.com/, accessed on 10 October 2021). ScienceDirect and WOS contain prestigious and high-quality journals. Full versions of the computer science-related articles of these journals are freely available to the authors of the paper in the majority of cases. These databases are used for reviewing papers that mention ontology learning tools while solving specific problems. Two search criteria are specified for searching in the ScienceDirect: (a) search terms: ("ontology learning" or "knowledge extraction") and tool, (b) considered time scale: 2016–2021. Search query in case of the WOS has the following structure where the same search terms and time scale is considered: ((TS = ("ontology learning") OR TS = ("knowledge extraction") AND TS = ("tool")) AND (PY = ("2021" OR "2020" OR "2019" OR "2018" OR "2017" OR "2016") AND DT = ("article" OR "review") AND SJ = ("computer science") AND OA = ("open access"))).

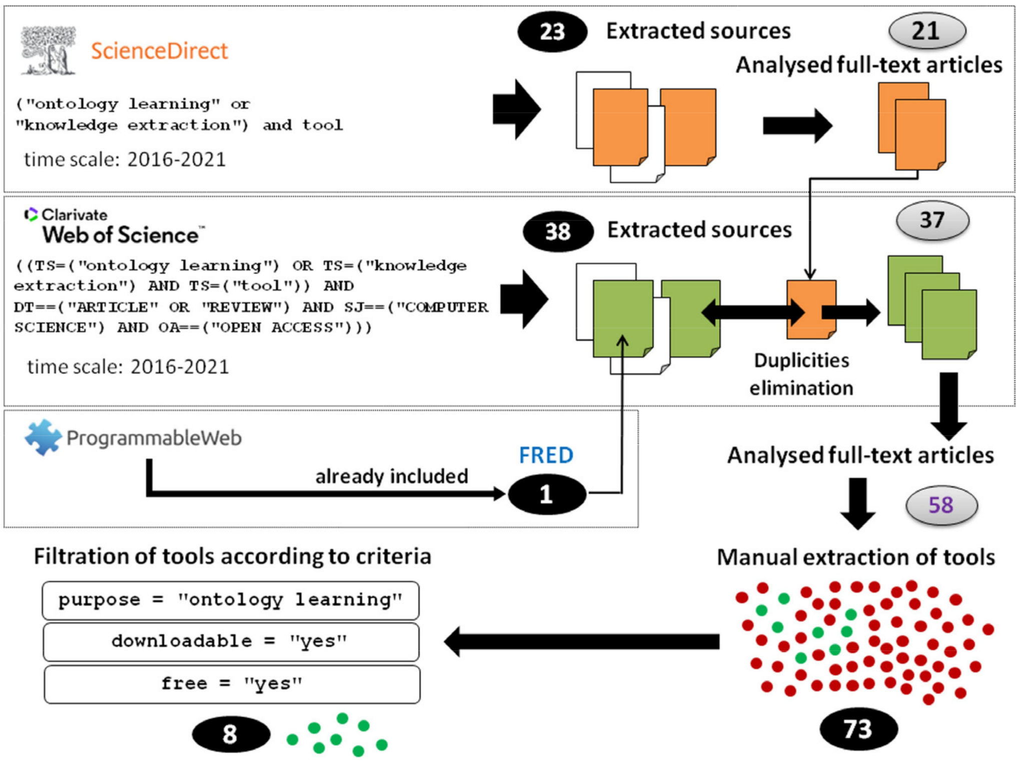

ScienceDirect
("ontology learning" or "knowledge extraction") and tool
time scale: 2016-2021
23
Extracted sources
21
Analysed full-text articles
Clarivate
Web of Science
((TS=("ontology learning") OR TS=("knowledge extraction") AND TS=("tool")) AND DT==("ARTICLE" OR "REVIEW") AND SJ==("COMPUTER SCIENCE") AND OA==("OPEN ACCESS")))
time scale: 2016-2021
38
Extracted sources
37
Duplicities elimination
ProgrammableWeb
already included
FRED
1
Analysed full-text articles
58
Manual extraction of tools
73
Filtration of tools according to criteria
purpose = "ontology learning"
downloadable = "yes"
free = "yes"
8

Figure 2. Illustration of the articles and tools selection process.

ScienceDirect returned twenty-three documents (see the black circle in Figure 2). Twenty-one documents represent journal articles, two of them book sections. Only documents having available full-texts were considered because these can provide all details necessary for our review. The search, in the case of the WOS, has returned thirty-eight documents (see the black circle in Figure 2). One duplicate document was detected during the comparison of results from ScienceDirect and WOS (see the grey circle in Figure 2). The final collection of documents used for a thorough analysis consists of 58 journal articles (see the grey circle with a violet number in Figure 2). Review data and studies together with the identification of tested tools are available in the Supplementary Materials.

Two categories of ontology learning tools were found in the journal articles: (a) ontology learning tools, which are directly used in solving a specific problem and thoroughly explained, (b) ontology learning tools, which are only mentioned in the paper, e.g., in the review paper or in the state of the art section of the paper. On the basis of analysis of the fifty-eight journal articles, statistics providing deeper insight into the used tools are mentioned in Section 4.1. Seventy-three tools are manually extracted from the 58 journal articles (see the black circle in Figure 2).

Three of the most important criteria were explored for answering the question of whether all of these tools can be practically used for the ontology learning. These criteria are visible in the Figure 2, in the Tables 4 and 5, and are the following:

(a) free solution available (no trial version)
(b) the ontology learning tool has to be downloadable and usable without and limitations
(c) the tool has to provide (semi-)automatic ontology learning

The ProgrammableWeb website is the third source that was used for searching the ontology learning tools. Almost the same search criteria were used in the exploration of the ProgrammableWeb. Downloadability and time scale cannot be specified on the website. No results were reached when the term “ontology learning” was used. One single result related to the FRED tool [19] was returned when the term “knowledge extraction” was used. This tool was included in the results returned by the search in the ScienceDirect and WOS database (see the black circle in Figure 2). The final subset of ontology learning tools consists of eight solutions from the total set of seventy-three tools (see the black circle in Figure 2). These ontology learning tools satisfy the above mentioned three criteria, which are predefined before deeper exploration of these tools. These criteria are also visible in Figure 2. Facts about seventy-three tools are completed on the basis of the ScienceDirect, WOS, and ProgrammableWeb search, i.e., used articles and homepages of tools (if they are available), see Tables 2–5. Facts about eight deeply analysed tools are completed similarly as in the previous case, together with the experimentation, with these tools, see Tables 6–8.

Table 2. Main purpose of the tool (tools 1–35).

Tool	**Data Aquisition**	**Extraction of Relationships**	**Ontology Development**	**Ontology Learning**	**Ontology Population**	**Ontology Visualisation**	**Reasoning**	**Recommendations**	**Searching**	**Streaming**	**Text Annotation**	**Text Mining**
AIDA											x	x
Alchemy API												x
Altova SemanticWorks			x									

Table 2. *Cont.*

Tool	Data Aquisition	Extraction of Relationships	Ontology Development	Ontology Learning	Ontology Population	Ontology Visualisation	Reasoning	Recommendations	Searching	Streaming	Text Annotation	Text Mining
Apache Jena			x									
Apache Stanbol							x		x		x	x
Apache UIMA				x								x
BRAT											x	
Caméléon					x							
CiceroLite											x	x
ClausIE		x										
DBPedia Spotlight												x
DeepKE		x										
DOE			x									
DOODLE				x								
FOX				x							x	x
FRED				x								
GATE					x						x	x
GEDITERM				x								
HOLMES					x							
HOZO			x									
ISOLDE		x		x								
Jaguar-KAT (JaguarTM)				x								
Jambalaya						x						
LinkFactory			x									
LODr											x	
MetaMap Lite											x	x
NERD											x	x
NooJ											x	x
OilEd			x									
Ollie		x										
OntoBuilder				x								
OntoCmaps				x								
OntoEdit			x									
Ontofier		x		x								
OntoLearn				x								

Table 3. Main purpose of the tool (tools 36–73).

Tool	Data Aquisition	Extraction of Relationships	Ontology Development	Ontology Learning	Ontology Population	Ontology Visualisation	Reasoning	Recommendations	Searching	Streaming	Text Annotation	Text Mining
OntoLiFT				x								
Ontolingua			x									
Ontology Learning Tool				x								
OntoLT				x								
OntoPOP					x						x	
Ontosaurus			x									
Ontotext												x
OntoX												x
OpenCalais											x	
OpenIE 4.0		x										
OWLExporter					x							
Pellet							x					
PoolParty KD									x		x	x
Protégé			x									
Protégé-OWL API			x									
RapidMiner												x
ReVerb		x										
SEISD				x								
Semiosearch (Wikifier)									x		x	x
SOBA				x								x
sProUT				x								x
Stanford CoreNLP												x
TaxGen				x								
TERMINAE				x								
Text2Onto				x								
TextRunner		x										
Text-To-Onto				x								
TopBraid Composer			x									
TRIPS												x
Twitter Streaming API										x		
Twitter's API	x											
Twitter´s Search API									x			
WebODE			x									

Table 3. *Cont.*

Tool	Data Aquisition	Extraction of Relationships	Ontology Development	Ontology Learning	Ontology Population	Ontology Visualisation	Reasoning	Recommendations	Searching	Streaming	Text Annotation	Text Mining
Welkin					x							
Wikimeta											x	
Wikipedia Miner	x											
WOE		x										
Zemanta								x				

Table 4. Selected characteristics of the tool (tools 1–35).

Tool	Directly Used in	Mentioned in	Downloadable	Free Solution
AIDA		[19]	yes	free
Alchemy API		[19,20]	no	N/A
Altova SemanticWorks		[21]	no	N/A
Apache Jena	[22]	[20]	yes	free
Apache Stanbol		[19,23]	yes	free
Apache UIMA		[24]	yes	free
BRAT	[25]		yes	free
Caméléon		[14]	no	free
CiceroLite		[19]	no	free
ClausIE		[26]	yes	free
DBPedia Spotlight	[27]	[19,20]	yes	free
DeepKE	[28]		no	free
DOE		[14]	no	free
DOODLE		[29]	yes	free
FOX		[19,23]	yes	free
FRED	[19,23,30]	[22]	yes	free
GATE	[22]	[24,29,31,32]	yes	free
GEDITERM		[14]	no	free
HOLMES	[31]		yes	free
HOZO		[33]	yes	free
ISOLDE		[34]	no	N/A
Jaguar-KAT		[35]	on request	N/A
Jambalaya	[36]		yes	free
LinkFactory		[21]	no	N/A

Table 4. *Cont.*

Tool	Directly Used in	Mentioned in	Downloadable	Free Solution
LODr		[27]	no	free
MetaMap Lite	[37]		yes	free
NERD		[19,23]	no	free
NooJ	[38]		yes	free
OilEd		[21]	yes	free
Ollie		[26,31]	yes	free
OntoBuilder		[29]	no	free
OntoCmaps		[26]	no	free
OntoEdit		[21,33]	no	N/A
Ontofier	[39]		no	free
OntoLearn		[14]	no	free

N/A: information not available free*: pricing policy includes a free plan trial: pricing policy includes trial version for free.

Table 5. Selected characteristics of the tool (tools 36–73).

Tool	Directly Used in	Mentioned in	Downloadable	Free Solution
OntoLiFT		[29]	no	free
Ontolingua		[21]	no	free
Ontology Learning Tool		[14]	no	free
OntoLT		[22,34]	no	free
OntoPOP		[31]	no	free
Ontosaurus		[21]	no	N/A
Ontotext		[24]	no	free*
OntoX		[24]	no	free
OpenCalais		[19,20,27]	yes	free
OpenIE 4.0		[26]	yes	free
OWLExporter		[40]	yes	free
Pellet	[36,41]		yes	free
PoolParty KD		[19]	yes	free
Protégé	[14,42][24,29][43]	[21,33,40,44]	yes	free
Protégé-OWL API	[22]	[20]	yes	free
RapidMiner		[31]	yes	free*, trial
ReVerb		[19,26]	yes	free
SEISD		[29]	no	free
Semiosearch (Wikifier)		[19]	yes	free
SOBA		[24]	no	free
sProUT		[24]	on request	free
Stanford CoreNLP	[27]	[30]	yes	free
TaxGen		[45]	no	N/A

Table 5. *Cont.*

Tool	Directly Used in	Mentioned in	Downloadable	Free Solution
TERMINAE		[14]	no	free
Text2Onto	[46]	[40,45]	yes	free
TextRunner		[26]	no	free
Text-To-Onto		[14,24,34]	yes	free
TopBraid Composer		[27]	yes	free*, trial
TRIPS		[24]	no	free
Twitter Steaming API	[20]		on request	free*
Twitter´s API	[20]		on request	free*
Twitter´s Search API	[20]		on request	free*
WebODE		[21,33]	yes	free
Welkin		[14]	no	free
Wikimeta		[19]	no	N/A
Wikipedia Miner		[20]	no	free
WOE		[26]	no	free
Zemanta		[19,27]	yes	free*

N/A: information not available free*: pricing policy includes a free plan trial: pricing policy includes trial version for free.

Table 6. Main features of the tool.

Tool	Distribution	Instant Download	Operability	Type of Tool	Active Development/Last Update	Degree of Automation	Supported Language	Documentation	Ease of Use	Installation
Apache UIMA	free	yes	yes	desktop (SDK)	yes/2021	semi	Java, C++ (Pearl, Python, TCL)	good	poor	hard
DOODLE-OWL	free	yes	restricted	desktop	no/2015	semi	N/A	poor	good	easy
FOX	free	yes	yes	Webservices, API	yes/2020	full	Java, Python	poor	good	easy
FRED	free	yes	yes	Webservice, API	yes/N/A	full	Python	poor	good	easy
OntoLearn	free	no	no	N/A	N/A	full	N/A	poor	N/A	failed
sProUT	N/A	no	N/A	N/A	no/2005	N/A	Java	poor	N/A	failed
Text2Onto	free	yes	no	desktop	no/2009	semi, full	N/A	poor	N/A	failed
Text-To-Onto	free	yes	restricted	desktop	no/2004	semi, full	N/A	poor	good	easy

Table 7. Functionalities explored in the evaluation.

Tool	Batch Mode Processing of Documents	Classes Extraction	Individuals (Instances) Extraction	Taxonomic Relations Induction (Concept Hierarchy)	Non-Taxonomic Relations Induction	Word-Sense Disambiguation	Coreference Resolution	Entity Linking
Apache UIMA	yes	partially [1]	yes [3]	N/A	N/A	N/A	N/A	N/A
DOODLE-OWL	yes	no	no	no	no	no	no	no
FOX	no	indirectly [2]	yes [3]	no	partially [4]	yes	N/A	yes
FRED	no	yes	yes	yes	yes	yes	yes	yes
OntoLearn	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
sProUT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Text2Onto	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Text-To-Onto	yes	yes	yes	yes	yes	no	no	no

[1]: types of named entities [2]: used predefined classes [3]: named entities [4]: used predefined annotation properties.

3.2. Tools Evaluation

The parameters for testing the eight ontology learning tools are divided into three categories. The category named "Main characteristics (general overview)" includes the following criteria:

- Distribution (free/N/A (information not available)
- Instant download: whether the tool can be downloaded without any limitations, e.g., no registration or password setting is required (yes/no)
- Operability: whether the tool can be run with no difficulties (yes/no/restricted/N/A)
- Type of tool: how the tool can be used by the end user (desktop/web service/API/N/A)
- Active development/last update: whether the tool is up-to-date and still actively developed (yes/no/N/A/last update)
- Degree of automation: how the text is processed by the tool (fully automatic/semi-automated/N/A)
- Supported programming language: whether it is possible to develop one's own ontology learning-based solution (programming languages)
- Documentation: how much the documentation is complete, correct, and comprehensible (pure/good)
- Ease of use: how easy it is to use the tool (poor/good/N/A)
- Installation: how easy it is to install the tool (easy/hard/failed)

It is clear that the parameters regarding the quality of documentation, ease of use, and ease of installation are rather subjective. However, these three parameters are crucial for tool usage and they can influence a decision whether the user will install the software or what he can expect from the tool.

Table 8. Supported formats of input/output.

Tool	**Supported Input Format**	**Supported Output Format**	**Graph-Based Visualisation of the Output**
Apache UIMA	txt	html, xml	N/A
DOODLE-OWL	txt	owl	no
FOX	txt, html, url	text/turtle, rdf/xml, rdf/json, json-Id, trig, n-quads	no
FRED	txt (string values)	rdf/xml, text/turtle, rdf/json, n3, nt, png, dag	yes
OntoLearn	N/A	N/A	N/A
sProUT	N/A	N/A	N/A
Text2Onto	N/A	N/A	N/A
Text-To-Onto	txt, hrml, xml, pdf	rdf	yes

The category named "Functionalities" covers functional aspects of the tool related to the aim of ontology learning. The most important functionalities of the tool are taken into account where the majority of them corresponds to the Ontology Learning Layer Cake.

An overview of these parameters is following:

- Batch mode processing of documents: whether more than one document can be processed in one step or whether they are processed one by one
- Classes extraction: whether classes as concepts can be extracted
- Individuals (instances) extraction: whether individuals, as instances of classes, can be extracted
- Taxonomic relations induction (concept hierarchy): whether it is possible to extract the subsumption hierarchy of concepts
- Non-taxonomic relations induction: this parameter corresponds to the extraction of non-taxonomic relations (i.e., domain-specific relationships)
- Word-sense disambiguation: whether the tool is able to detect the correct meaning of the word in a specific context
- Coreference resolution: whether it is possible to find all expressions referring to the same entity in the text
- Entity linking: whether the tool is able to recognize (named) entities in the text and assign a unique identity to their knowledge base counterparts

The third category of parameters covers supported input and output formats, and whether the tool provides graph-based visualisation of the results.

4. Results

4.1. Ontology Learning Tools: Analysis of Journal Articles

Three categories of journal articles are found in the collection of 58 journal articles: research articles, review articles, and comparative studies. These three groups can be subdivided into more specific categories: research articles where a concrete tool is used in solving a specific problem, research articles where a concrete method or technique is used (e.g., machine learning algorithm), articles providing a review of tools or review of methods (techniques) for semantics extraction (ontology learning), comparative studies, and other(s) articles mainly presenting challenges or future perspectives related with the processing of big data. It was not a surprise that more tools (not only focused on the ontology learning) were found. This fact arises from the nature of the query, which has been used in searching in the ScienceDirect and WOS. Seventy-three tools for ontology learning or knowledge extraction are found. Tables 2 and 3 provide the main purpose of these tools, and Tables 4 and 5 show they can be currently downloaded and used.

Figure 3 presents how many tools are mentioned in the articles with respect to the main purpose. It is visible that the first three positions belong to the solutions used for ontology learning (21), text mining (20), and text annotation (15).

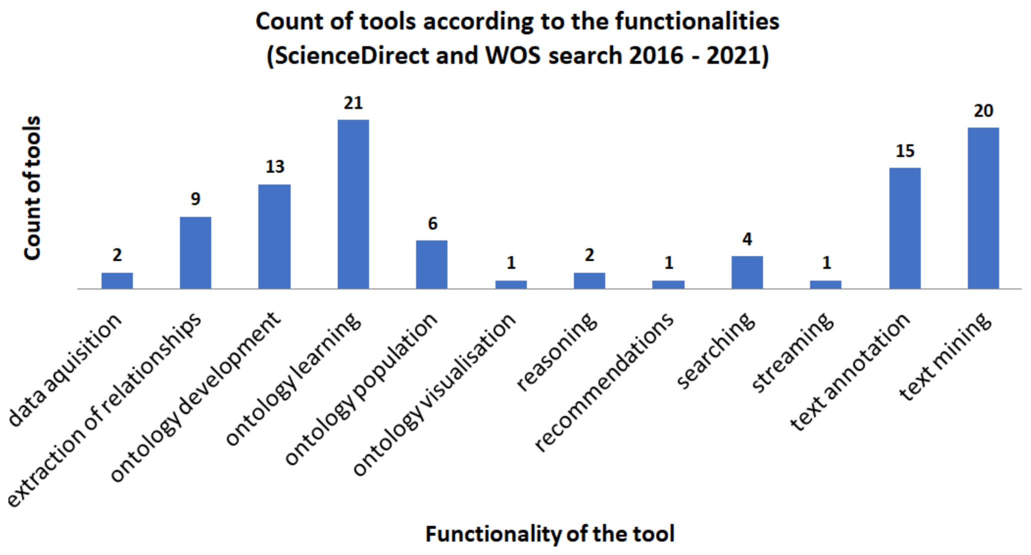


Figure 3. Count of tools according to their functionalities—ScienceDirect and WOS search (2016–2021).

Three of the most important aspects are followed and used to filter relevant tools: (1) purpose of the tool, (2) whether the tool can be instantly downloaded and (3) whether the tool is freely available. There are 46% of tools that can be instantly downloaded. There are 47% of tools that cannot be instantly downloaded, but 7% of tools can be downloaded on request. There are 76% of tools that are freely available. Additionally, 9% of tools provide a free plan, and 3% of tools provide a trial version. Information about price politics is not available for 12% of tools.

The count of occurrences of these solutions in the journal papers can be deduced from Tables 4 and 5. It is visible that Protégé is the most cited tool. GATE text engineering platform has the second-highest score, but this one is mainly used for text analysis. Ontology population can be realized with this platform. FRED machine reader has the third-highest score. This tool is mainly used for ontology learning.

If we focus more on the ontology learning category, where 21 tools are found, 29% of the OLTs (Ontology Learning Tools) are instantly downloadable. There are 62% of OLTs that are not instantly downloadable, while 9% of OLTs can be received on request, and 86% of OLTs are freely available. Information about prices politics is not found for 14% of OLTs. No OLT is provided as the trial version. Based on the three basic requirements for the ontology learning tools, which are mentioned above, eight ontology learning tools are selected for deeper analysis: Apache UIMA, DOODLE-OWL, FOX, FRED, OntoLearn, sProUT, Text2Onto, and Text-To-Onto (see Table 6).

4.2. Evaluation of Ontology Learning Tools

As it is mentioned in the introductory section, the primary motivation behind this paper is to automatically process the unstructured texts, which are mainly focused on the tsunami research, more generally on disaster management. As a brief explanation, tsunamis are generally considered as high impact but low-frequency natural events. Tsunami waves are generated mainly from submarine earthquakes, but they can also be generated from other types of sources, including volcanic eruptions, landslides, and meteorological changes (meteotsunamis). In the open sea, the tsunamis are treated as solitons, whereas in the shallow water domain, they are train waves [47]. Not all tsunamis are damaging in the coastal zones. However, strong tsunamis that hit populated coastal areas may cause various types of impact. In the built environment, the tsunami waves may cause substantial damage in vessels, infrastructures, buildings, and other properties. In the natural environment, tsunamis cause coastal destruction and erosion. Big events may also cause loss of human life and injuries. Preparedness actions are of high importance to reduce tsunami risk [48]. To this aim, preparedness actions may include the elaboration of emergency plans, education, and training, as well as development of early warning systems.

The abstract of the article published in Natural Hazards journal in 2018 [49] is used as the input for investigation of the tools for ontology learning. This article describes the tsunami, which occurred in 2017 on the southern coastlines of Iran (Bandar Dayyer country), resulting in significant financial losses, damage to the port, missing people, and one death. This tsunami is perceived as meteotsunami by the authors because of the atmospheric disturbances which occurred before this event. The authors use the simulations for a better understanding of this disaster. The following short piece of text is used for tools analysis:

> *"We present a field survey and a number of simulations of the local Persian Gulf tsunami of 19 March 2017 at Bandar Dayyer, Iran, which resulted in one death, five persons missing and significant damage to the port. The field survey defined the inundated area as extending 40 km along the coast, with major effects concentrated on an 8 km stretch immediately west of Dayyer, a maximum run-up of 3 m and maximum inundation reaching 600 m. In the absence of significant earthquakes on that day, we first test the possibility of generation of a landslide; however, our simulations for legitimate sources fail to reproduce the distribution of run-up along the coast. We prefer the model of a meteorological tsunami, triggered by Proudman resonance with a hypothetical weather front moving at 10 m/s in a NNW azimuth, which could be an ancillary phenomenon to a major shamal wind system present over the Persian Gulf on that day. More detailed simulations of the Dayyer tsunami would require an improved bathymetric grid in the vicinity of the relevant coastal segment."*

4.2.1. Text-To-Onto

Text-To-Onto (TextToOnto) [50] is a freely available tool supporting ontology engineering-based activities using the text mining techniques. Text-To-Onto is based on the ontological infrastructure named KAON (Karlsruhe Ontology). It supports management of the RDF (Resource Description Framework) [51] ontologies. It was developed by the University of Karlsruhe and the Research Center for Information Technologies in Karlsruhe in 2002. The authors of the paper verify that it is possible to build the ontology in the OI-modeller manually without any problems. The ontology is saved as the *.kaon file. Source code of the *.kaon file corresponds to the RDF formal ontology, which can be opened in an RDF-based ontological editor, e.g., in the TopBraid Composer [52]. Text-To-Onto accepts txt, html, xml, and pdf files for corpus building. More files can be a part of one corpus, which cannot be saved for future usage. Eight additional functionalities are promised by the tool:

- extraction of terms

- extraction of association (without domain/range)
- ontology pruning (automated extraction of irrelevant concepts to the application domain where the ontology generalizes the target domain)
- taxonomy induction
- extraction of instances
- relations learning (including domain/range)
- ontology enrichment (ontology extension with additional concepts/relations)
- ontologies comparison

New term extraction seems problematic where no results are provided by the tool. The same feedback is given with and without ontology. In the case of taxonomy induction, new is-a relationships are detected using the Formal Concept Analysis (FCA). This is based on the detection of the most frequent terms in the corpus or terms detected during the terms extraction process. No results are given when a combination-based approach is applied. This approach uses a mixture of information sources for is-a relationships detection, i.e., WordNet lexical database [53]. The changes are also immediately included in the ontology without their saving. This is unexpected behaviour. Sometimes, the process of taxonomy induction is incomplete because the program stops to work and does not give any response. The impossibility to save preferences in the tool is another unexpected behaviour. From a subjective point of view, it is not easy to find the place where to find results of ontology learning. More details about testing can be found in Table 7.

4.2.2. Text2Onto

Text2Onto is a well-known framework used for ontology learning from the unstructured texts [54,55]. Linguistics-based operations (e.g., tokenization, lemmatization, stemming, and shallow parsing) and machine learning algorithms are combined for (semi-) automated extraction of ontological structures. Text2Onto is the re-designed version of the Text-To-Onto [50]. Extracted ontology is represented by so called “Modelling Primitives” which are then transformed into specific modelling language, e.g., RDFS (Resource Description Framework Schema) [56], OWL (Ontology Web Language) [57], or F-Logic [58,59]. This ensures that the output of the ontology learning is language independent, and it is possible to avoid often problematic transformations between different formalisms. Text2Onto distinguishes the following modelling primitives [54]:

- concepts
- taxonomical relations
- concept instantiation
- properties
- domain and range restrictions
- mereological relations, i.e., part-whole relationships
- equivalence

Probabilistic ontology model (POM) is a collection of instantiated modelling primitives. Text2Onto also provides a mechanism called data-driven change discovery, thanks to which it is not necessary to process the corpus each time when it is changed by the user. The probabilistic ontology model is used for this purpose: when the corpus is changed, only the probabilistic ontology model is updated according to these changes.

Text2Onto can be installed in two ways. The first one depends on the installation of the archive file [60]. The second one is based on the NeOn ontological editor using the Text2Onto plugin [61]. Both approaches lead to installation problems. Text2Onto relies on the GATE 4.0 text engineering platform [62] and WordNet 2.0 [53,63]. GATE 9.0.1 (Windows version released on March 2021) and WordNet 2.1 (Windows version released on March 2005) are the latest versions at present. The content of the specific Text2Onto files have to be set up in various places. Some of these important settings are not mentioned in the instructions files of the Text2Onto, but they are documented in various threads used for solving problems [64,65]. Installation of the Text2Onto finally fails.

In the case of the second way, only the NeOn toolkit in version 1.x supports the Text2Onto plugin. This version cannot be downloaded because the link for downloading is broken. A similar experience is reported in [46]. The Text2Onto was used for ontology extraction from unstructured documents, which are focused on laundry detergent product criteria. An unsatisfactory conclusion about this tool is mentioned [46]. More details about testing can be found in Table 7.

4.2.3. FRED

FRED is a freely available so-called “machine reader for the Semantic Web” [66]. “Machine reader” is a tool that is able to transform the unstructured text into a formal structure—the ontology [67]. FRED is able to interconnect the automatically generated formal structure with already available pieces of information and knowledge, e.g., with the facts available in the DBPedia knowledge base [68], WordNet lexical database for English language [53] or schema.org vocabulary [69]. This is realized especially by the ontology alignment or entity linking techniques [19]. FRED intensively draws from the NLP techniques, including named entities recognition, coreference resolution, or word sense disambiguation. In the view of (formal) linguistics, FRED is based on the combinatory categorical grammar [70], discourse representation theory [71,72], and frame semantics [66,73,74]. FRED is available as the HTTP REST service or Python library. Online version for experimentation is also available on the web [75]. The online version of the FRED has been tested. In this version, various settings are available [75]. The user can set up the following parameters in the FRED online version (values of the parameters used during the testing are given in the square brackets []):

Namespace prefix for FRED terms [fred:]

- Namespace for FRED terms [http://www.ontologydesignpatterns.org/ont/fred/domain.owl] (accessed on 15 September 2021)
- Do word-sense disambiguation [yes]
- Align concepts to Framester [no]
- Tense [yes]
- Use FrameNet roles [no]
- Always merge discourse referents in case of coreference [no]
- Text annotation format [EARMARK]
- Return the semantic-subgraph only [no]

Processing of documents in a batch mode is not accessible in the online version or in the HTTP REST service, but a programmer can ensure batch processing on own Python-based program. FRED accepts unstructured texts and transforms them into various formats (see Table 8).

The unstructured text was sent into the FRED for processing, and the Turtle file was produced. FRED also provides the graph-based visualisation of the result, but if a text is complex, this visualisation is hardly readable. This is the reason why the Turtle file was later opened in the Protégé 5.2.0, where the OntoGraf plugin was used for customized visualisation. The OntoGraf depicts ontological classes, individuals, and relationships between them. The user can select which part of the ontology should be visible. Taxonomic and non-taxonomic (object, annotation) relationships were induced together with the individuals (instances of ontological classes). Entity linking is ensured by the DBPedia (example: dbpedia:Tsunami isEquivalentTo Tsunami (extracted concept by FRED)) and schema.org (example: dbpedia:Iran isInstanceOf schema:Place).

Automatically generated formal ontology is also enriched by so-called synsets. The synset is a set of one or more synonymous words. These words are attached to extracted concepts (classes) in the ontology. Part of speech is directly visible in the name of the synset (examples: Tsunami isEquivalentTo synset-tsunami-noun; Bathymetry isEquivalentTo synset-bathymetric-adjective). FRED is also able to detect synonymous words. It is visible, for example, for Inundation concept having Flood concept as its equivalent (example:

Inundation isEquivalentTo dbpedia:Flood isEquivalentTo synset-flood-noun). Due to these enrichments, the ontology is more complex and hardly readable.

Figure 4 depicts the fragment of the generated ontology, visualized by the OntoGraf, where the Dayyer individual various relationships are induced by the FRED. Figure 5 is focused on the Event class, where the various different events are extracted by the FRED.

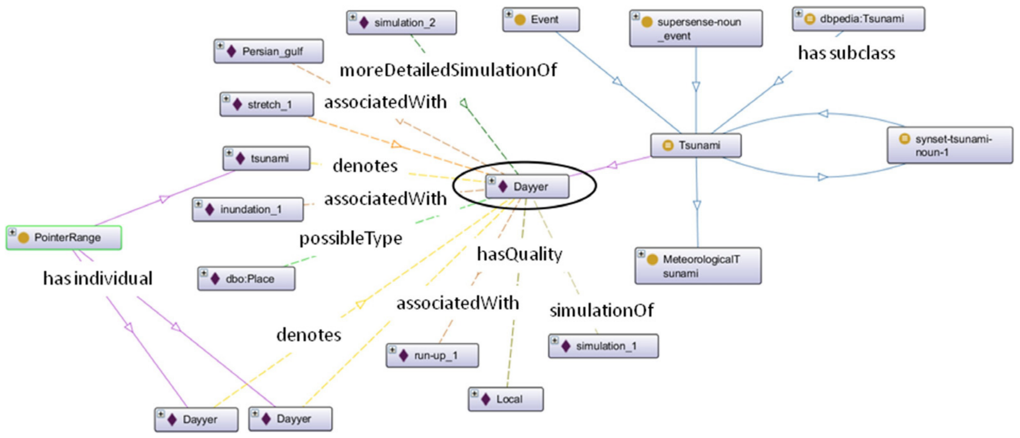


Figure 4. Fragment of the generated ontology (Dayyer in the circle).

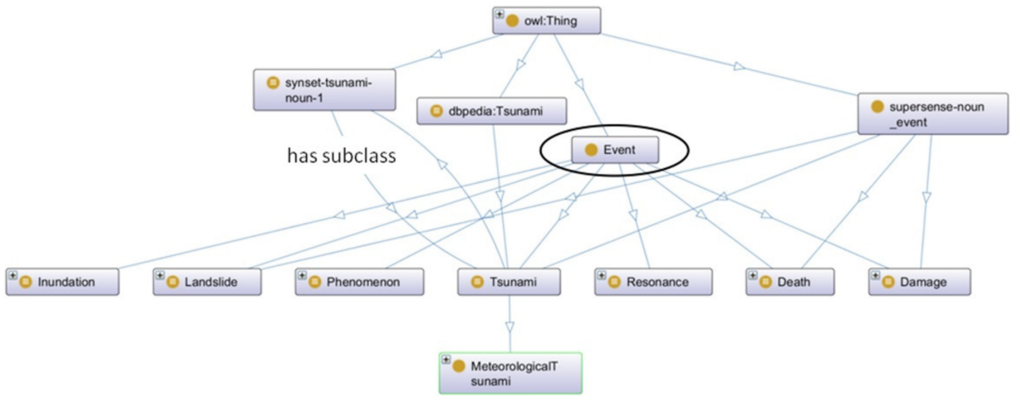


Figure 5. Fragment of the generated ontology (Event in the circle).

4.2.4. DOODLE-OWL

DOODLE-OWL (a Domain Ontology rapiD DeveLopment Environment—OWL extension) is freely available Java-based development environment for semi-automatic building of the OWL ontologies [76]. The OWL (Ontology Web Language) is the W3C standard used for building ontologies based on description logic [57,77,78]. These highly formal structures are suitable for expression semantics in the semantic web context. DOODLE-OWL is a redesigned version of the DOODLE II [79] and DOODLE [80], which are not customized for the Semantic Web-based environment. Authors of the DOODLE-OWL declare the following six modules used for interactive development of domain OWL ontologies:

- Input module is used for selecting the inputs, which are fundamental for ontology learning, especially: WordNet [53], ERD [81], text documents written in English or Japanese, and the list of extracted terms (concepts) from an unstructured text.
- Construction module is responsible for the extraction of the taxonomical (is-a) and the non-taxonomical relations.
- Refinement module is inside the construction module. It identifies significant pairs of concepts in the extracted related pairs of concepts. This is realized interactively with the user.

- Visualisation module is represented by the RDF(S)-based graphical editor, providing visualisation of the ontological structure, including consistency checking of the ontological classes.
- Translation module exports generated structure into the OWL format.

ActiveState Perl is necessary for English and Japanese written texts. ActivePerl 5.28 is used for experimentation with the DOODLE-OWL. The correct path to the Perl has to be setup in the DOODLE-OWL, including project folder, the list of stopwords, and ERD (if necessary). Input module provides WordNet in version 3.0 and 3.1 for setting, but Wordnet 2.1 is the most actual Windows version for downloading. WordNet 3.0 is available for Linux and Mac-OS X. WordNet 3.1 does not contain the code for running WordNet. It can be speculated that, maybe, this is the reason why no taxonomical and non-taxonomical relationships are extracted in DOODLE-OWL. Documentation for DOODLE-OWL declares that the DOODLE-OWL has only been tested on the macOS Sierra operating system [82].

Three taggers are used for words extraction. Gensen extracts complex words written in English and Japanese [83]. Sen is used for the extraction of Japanese words and their parts of speech [76]. SS-Tagger extracts English words and identifies their parts of speech [84]. DOODLE-OWL attaches corresponding parts of speech for all words, but the results are not visible in the GUI but, rather, in a separated file, which is automatically generated by DOODLE-OWL "behind the scene". Experimentation shows that only parts of speech tagging is provided by DOODLE-OWL, where the 64-bit system with the OS Windows 10 and Java version 1.8.0 (281) is used.

Documentation for DOODLE-OWL is pure. It would be useful to provide two types of documentations. The first one is only for the English audience, and the second one for the Japanese audience. Actual documentation is a mixture of both of them. More details about testing can be found in Table 7.

4.2.5. OntoLearn

OntoLearn is a system developed for taxonomy induction from unstructured texts using text mining and machine learning techniques [85]. Already developed initial ontology has to exist. This ontology is enriched and more concretized (trimmed), according to the domain texts, which are part of the corpus. OntoLearn mainly uses WordNet [53] as the generic ontology. It means that OntoLearn is mainly focused on the ontology population.

OntoLearn occurred in the context of the two EU projects: Harmonize (2001–2003) and INTEROP (2003–2007). The main aim of the Harmonize was to provide means for ensuring technological interoperability and implementation of cooperation models for software and business processes focused on tourism. Reconciliation among tourism standards with the usage of formal ontologies was also a part of this project. INTEROP was aimed at re-structuring the area of enterprise interoperability research in Europe.

OntoLearn system is based on the three fundamental steps [86]:

1. terminology extraction from corpus
2. semantical interpretation of extracted terms
3. arrangement of these terms into the hierarchy

Terminology extraction uses WordNet lexical database, but others can also be used for this purpose. SSI (Structural Semantic Interconnections) algorithm is used during the second phase [86]. It is the word sense disambiguation algorithm that determines correct sense of words (multi-word expressions). The OntoLearn system consists of algorithms for terminology extraction, sense disambiguation, and extraction of semantic relations. Resulted ontology is encoded in the OWL format.

OntoLearn Reloaded is the extended version of OntoLearn [85]. It is a graph-based algorithm also used for taxonomy induction where WordNet is not used because of the dependence on the English language. SSI algorithm is also not used in this version. The philosophy behind the OntoLearn Reloaded is to build automatically the Directed Acyclic Graphs (DAG) using the optimal branching algorithm.

In view of the practical usage of the above-mentioned solutions, neither of them can be downloaded and used because no site for downloading is not reachable. Web support for OntoLearn Reloaded exists (http://www.ontolearn.org, accessed on 15 September 2021), but it mainly contains results received from OntoLearn Reloaded. For avoiding confusion, the Ontolearn is accessible on the web, but it is the Python-based open-source software library used for concept learning on the RDF knowledge bases [87]. More details about testing can be found in Table 7.

4.2.6. Apache UIMA

Apache UIMA (Unstructured Information Management Architecture) is the Apache-licensed open-source reference implementation of the UIMA specification [88], originally developed at the IBM. The UIMA specification is the OASIS standard defining platform-independent data representation and interfaces for processing of the unstructured textual, video and audio documents using specialized software components or services. The UIMA supports interoperability among different platforms or frameworks which, already exist.

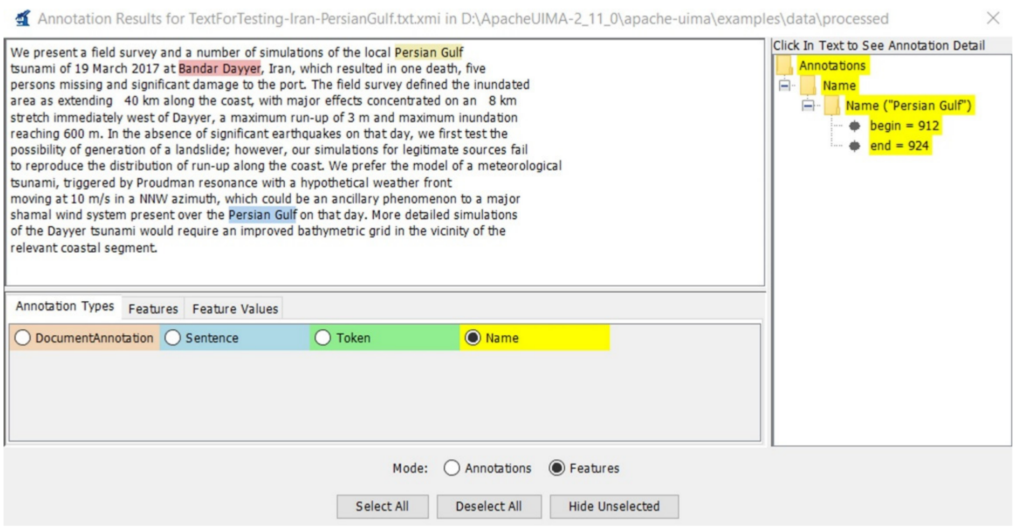
The main motivation behind the UIMA is to provide an easily extendable solution for analysis of unstructured documents, where components of the solution can be reused. Reusability is realised by the plugin-based architecture. These reusable components are managed by the UIMA framework ensuring data flow between them. The Apache UIMA consists of several components where each one is responsible for particular task. Large collection of components is dedicated to the annotators, e.g., Whitespace Tokenizer Annotator, Snowball Annotators using the Snowball stemming algorithm, Regular Expression Annotator using the regular expressions for annotation or Dictionary Annotator using the list of words compiled into the simple dictionaries.

Apache UIMA provides the Apache UIMA sandbox—a workspace for accessing own piece of code to other UIMA developers. Various built-in tools are provided by the Apache UIMA. Document Analyzer is a tool that can be used for testing the annotators. UIMA Ruta (Rule-based Text Annotation) is a rule-based system used for information extraction [89]. It uses a rule-based language utilizing regular expressions for information extraction. The UIMA Ruta Eclipse-based Workbench is used for rules modelling. The Annotation viewer can apply annotation techniques for its own texts, and it visualises the results of annotations. Unstructured text about tsunami event at Bandar Dayyer country (Iran) is annotated by this viewer, see Figure 6.

Apache UIMA provides framework implementation in Java (Apache UIMA Java framework) and C++ (Apache UIMA C++ framework) programming language. The C++ framework also supports annotators written in Perl, Python and TCL. The Apache UIMA promises the following techniques (approaches), namely [90]:

- statistical techniques
- rule-based techniques
- information retrieval
- machine learning
- ontologies
- automated reasoning
- integration of knowledge sources (e.g., WordNet)

These technologies (techniques) are used for the transformation of the unstructured documents into structured form. Extracted structures are then used by conventional technologies, e.g., search engines, database engines, OLAP (On-Line Analytical Processing) engines and access the structured content to the user [91]. Various solutions use the Apache UIMA, e.g., Apache OpenNLP [92], INCEpTION [93], or JulieLAB NLP Toolsuite [94]. Developers can choose already developed suitable UIMA components or to use own components and connect them into the functioning application. Fully-fledged usage of the Apache UIMA framework depends on the programming abilities of the user. Usage of the Apache UIMA is mainly based on programming the annotators, which are able to add extra information to the tokens. Annotators can distinguish into which class the token belongs to. Is it a place, a number, a person, or a date? The user can use the built-in annotators where no programming experience is required, but these annotators are limited to the identification of specific named entities. Building of the own annotators are based on Java programming knowledge, XML and regular expressions. The Figure 6 depicts the result of the annotation of our text about tsunami where one of the built-in annotators is used in the Eclipse platform.


Annotation Results for TextForTesting-Iran-PersianGulf.txt.xmi in D:\ApacheUIMA-2_11_0\apache-uima\examples\data\processed

```
We present a field survey and a number of simulations of the local Persian Gulf
tsunami of 19 March 2017 at Bandar Dayyer, Iran, which resulted in one death, five
persons missing and significant damage to the port. The field survey defined the inundated
area as extending  40 km along the coast, with major effects concentrated on an  8 km
stretch immediately west of Dayyer, a maximum run-up of 3 m and maximum inundation
reaching 600 m. In the absence of significant earthquakes on that day, we first test the
possibility of generation of a landslide; however, our simulations for legitimate sources fail
to reproduce the distribution of run-up along the coast. We prefer the model of a meteorological
tsunami, triggered by Proudman resonance with a hypothetical weather front
moving at 10 m/s in a NNW azimuth, which could be an ancillary phenomenon to a major
shamal wind system present over the Persian Gulf on that day. More detailed simulations
of the Dayyer tsunami would require an improved bathymetric grid in the vicinity of the
relevant coastal segment.
```

Click In Text to See Annotation Detail

```
Annotations
  Name
    Name ("Persian Gulf")
      begin = 912
      end = 924
```

Annotation Types | Features | Feature Values

DocumentAnnotation [ ] Sentence [ ] Token [ ] [✓ Name]

Mode: Annotations [✓ Features]

Figure 6. Annotation of the unstructured text about meteotsunami in the Apache UIMA frame- work.

Identification of the named entities is close to the taxonomy induction, but it is far from the sophisticated ontology learning tool. No built-in plugin or the Apache UIMA component directly supporting ontology learning is not found in the Apache UIMA website [95]. Text mining-based techniques are used in ontology learning. Authors of the paper [91] introduce the systematic review of text mining-based solutions which utilizes the Apache UIMA. This review emphasizes that Apache UIMA is widely used in the information extraction and NLP, using the Annotation engine. No studies focusing on the usage of the Apache UIMA directly during ontology learning are not mentioned in this research. On the other hand, research studies that mention the usage of the Apache UIMA framework for ontology-learning related tasks are found. The authors of the paper [96] introduce a terminology development environment named Edtgar. Edtgar provides a module for terminology induction and refinement, using a corpus of plain text documents and Apache UIMA framework. The terminology induction is based on the detection of the part-of-speech tags, so especially nouns and adjectives are detected. Then, the lemmas of these words are found and used for their grouping. Words having the same lemmas are closest to each other. In this way, different variants of the same word are found. Authors of the paper [97] introduce the Java-based Term Extraction Tool, named as TermSuite, for automatic building of terminologies from a corpus using the NLP techniques (tokenization, parts of speech tagging, lemmatization, stemming, splitting) and Apache UIMA framework. The UIMA Tokens Regex component is used for spotting multiword terms. Apache UIMA framework attempts to transform annotated content (metadata) into RDF-based structures using the RDF UIMA CAS (Common Analysis Structure) Consumer, but its functionalities are limited [98]:

"...do not support projecting information towards specific RDF vocabularies."

CODA (Computer-aided Ontology Development Architecture) tries to fill this gap. It is the architecture and framework able to consume unstructured and semi-structured textual documents and transform them into the RDF datasets. CODA Analysis Engine is one of the main components responsible for the population of the RDF datasets. This solution extends the UIMA, where UIMA annotated content (metadata) is represented as

RDF triples [99]. As in the case of the Apache UIMA, usage of the CODA is dependent in programming background in Java and specific language, named as Pearl (ProjEction of Annotations Rule Language)—a rule-based pattern matching and transformation language responsible for the projection of the UIMA metadata into the RDF graph patterns [100]. More details about testing can be found in Table 7.

4.2.7. SProUT

SProUT (Shallow Processing with Unification and Typed feature structures) is the multi-purpose engine for various multilingual NLP tasks, including named entity recognition, information extraction, topic modelling, or ontology extraction from unstructured texts [101,102]. It is implemented in Java and C++. It supports the processing of unstructured texts in English, German, French, Italian, Dutch, Spanish, Polish, Czech, Chinese, and Japanese. Three main aims are specified for the engine:

1. to provide a system that integrates different modules for texts processing
2. to find a compromise between processing efficiency and expressiveness of the formalism

SProUT consists of several key components, namely a finite-state machine toolkit, a regular compiler, a finite-state machine interpreter, a typed feature structure package, and a collection of linguistic processing resources including a tokenizer, a gazetter, a morphology component, and a reference matcher [103]. These components are used, as a whole, for the development of the multilingual shallow text processing systems. It plays a role for content extraction, machine translation, or text summarization [103].

In the view of the experimentation with this tool, no possibility for downloading of the SProUT is found. More details about testing can be found in Table 7.

4.2.8. FOX

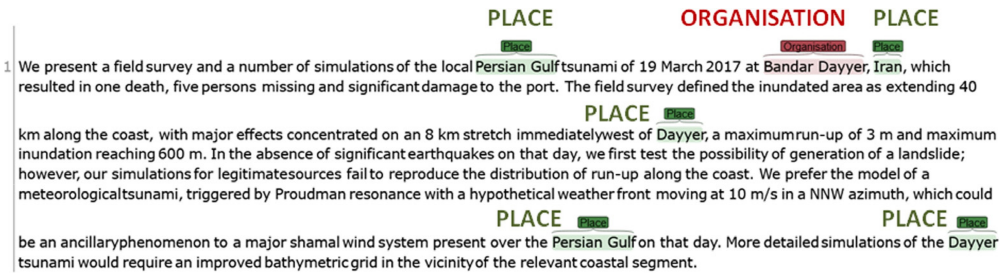
FOX (Federated knOwledge eXtraction) is the open-source framework mainly used for named entity recognition [104,105]. This framework is developed by the project team of the AKSW (Agile Knowledge Engineering and Semantic Web) research group of the Leipzig University. The reason why this tool is included in the list of tested tools is that FOX transforms the unstructured text into the RDF triples. FOX detects the named entities for the Location, Organization, and Person named entity type. It facilitates the open-source named entity disambiguation framework, named as AGDISTIS [106], for named entities disambiguation and links them to the DBPedia knowledge base. Entity linking is not limited only to the DBPedia.

It integrates four state-of-the-art named entity recognition frameworks, namely the Stanford Named Entity Recognizer [107], the Illinois Named Entity Tagger [108], the Ottawa Baseline Information Extraction [109], and the Apache OpenNLP Named Finder [92]. The user can choose and apply this recognizer for text processing in the FOX (with the FOX light option). FOX functions can be used on Java-based or Python-based applications.

Unstructured text about meteotsunami is also used for testing the FOX with the following settings in the demo version [105]:

- Lang (language): en
- Input Format: text/html
- Extraction Type: ner (named entity recognition)
- Input (unstructured text): document about meteotsunami
- Output Format: Turtle
- Fox Light: OFF

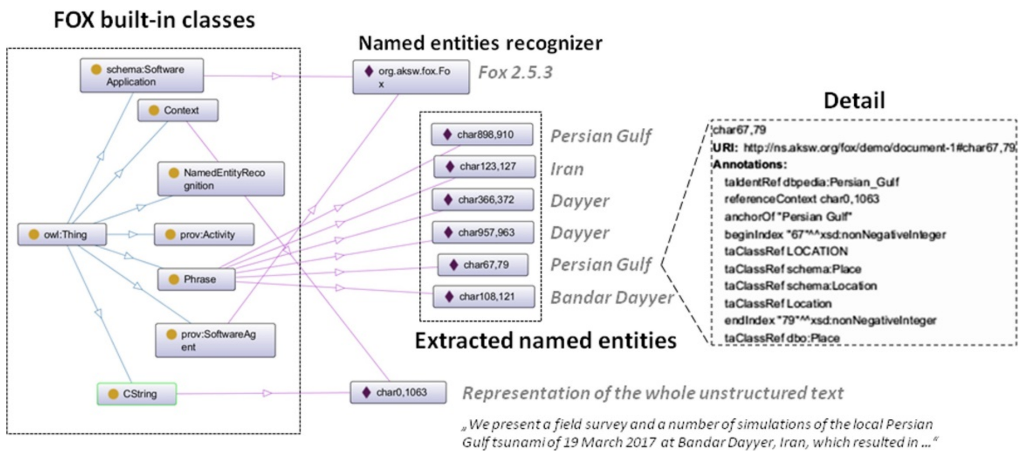
FOX provides a simple visualization of named entities with their types. FOX correctly detects five named entities, but in the case of the concept, named as Bandar Dayyer, it cannot be said that it is the organisation, see Figure 7.



We present a field survey and a number of simulations of the local Persian Gulf tsunami of 19 March 2017 at Bandar Dayyer, Iran, which resulted in one death, five persons missing and significant damage to the port. The field survey defined the inundated area as extending 40 km along the coast, with major effects concentrated on an 8 km stretch immediately west of Dayyer, a maximum run-up of 3 m and maximum inundation reaching 600 m. In the absence of significant earthquakes on that day, we first test the possibility of generation of a landslide; however, our simulations for legitimate sources fail to reproduce the distribution of run-up along the coast. We prefer the model of a meteorological tsunami, triggered by Proudman resonance with a hypothetical weather front moving at 10 m/s in a NNW azimuth, which could be an ancillary phenomenon to a major shamal wind system present over the Persian Gulf on that day. More detailed simulations of the Dayyer tsunami would require an improved bathymetric grid in the vicinity of the relevant coastal segment.

Figure 7. Annotation of the unstructured text about meteotsunami in the FOX framework.

The Turtle-based file can be opened in an ontology editor (e.g., in Protégé 5.2.0) with more details visible (see Figure 8). The resulting RDF file contains predefined ontological classes (Context, CSString, NamedEntityRecognition, Phrase, prov:Activity, prov:SoftwareAgent, schema:SoftwareApplication). Each one "contains" some resources (individuals)—named entities. Named entities of the named entities types (Location, Organization, Person) are "contained" in the Phrase class, see Figure 8. Various annotation properties are linked to these extracted resources, e.g., char67,79 (a resource) anchorOf (an annotation property) PersianGulf (extracted named entity, value of the annotation property), char67,79 (a resource) taIdentRef (an annotation property) http://dbpedia.org/resource/Persian_Gulf, accessed on 10 October 2021 (value of the annotation property) or char67,79 (a resource) taClassRef (an annotation property) http://schema.org/Place, accessed on 10 October 2021 (value of the annotation property).



```
char67,79
URI: http://ns.aksw.org/fox/demo/document-1#char67,79
Annotations:
   taIdentRef dbpedia:Persian_Gulf
   referenceContext char0,1063
   anchorOf "Persian Gulf"
   beginIndex "67"^^xsd:nonNegativeInteger
   taClassRef LOCATION
   taClassRef schema:Place
   taClassRef schema:Location
   taClassRef Location
   endIndex "79"^^xsd:nonNegativeInteger
   taClassRef dbo:Place
```

Figure 8. Visualisation of the resulted Fox-based RDF taxonomy in the Protégé 5.2.0.

It was expected that the named entity types will be directly modelled as the onto-logical classes and named entities as instances of these classes. On the basis of the above mentioned experience, it cannot be said that FOX provides a suitable tool for ontology learning or taxonomy induction, but it is a promising alternative for named entities recognition.

5. Discussion and Future Directions

Three categories of articles were found during exploration of the ScienceDirect scientific database: research articles (14), reviews (6), and the comparative article (1). One review article [110] is mainly focused on the elaboration of methods used for ontology learning. Rigorous knowledge systematization of the methods is realized by the OWL formal ontology, which models ontology learning methods. Only three reviews mention tools used for ontology learning. The first one [24] provides the state of the art for so-called ontology-based Information Extraction systems and their comparative analysis. Systematization of these ontology-based information extraction systems is also modelled by the OWL ontology. It is not clear whether these tools were directly tried (tested) by the author of the article for comparative analysis. It seems that comparative analysis is realized on the basis of available literature sources, which mention or describe these tools.

The second review article [14] presents and analyses various approaches for ontology learning. The knowledge repository containing these methods is represented by the OWL ontology. It also mentions a couple of solutions used for ontology learning, but these are not described in detail because this work extends the previous one [7], where a comprehensive review of tools for ontology learning is presented. The review is comprehensive, yet no details about the practical experience with these tools are presented. The author applied a different approach: to provide a review, as in the case of above mentioned papers.

The third review article [29] extends the previous one [110], where the OWL ontology representing the ontology learning approaches is extended and verified by the competence questions. Only a limited number of tools for ontology learning are mentioned in this article. Other reviews [35,111] are not directly focused on ontology learning and the tools.

Two categories of articles were found during the investigation of the WOS scientific database: research articles (36) and reviews (2). The systematic study of journal articles, available in the ScienceDirect and WOS (2016–2021), shows that no review mentions any experience with tools, for example: *"Is a tool available during writing the paper?"*, *"Is it possible to easily download or install a tool?"*, *"Which functions/methods are really available in a tool?"*, etc. The authors of the reviews often apply a different approach. This is acceptable, but the reader of these papers should be aware of additional facts. The reader may get an impression that all these tools can be used without any limitations. It is important to realize that computational platforms and requirements on these tools change over time. It is clear that we can expect software installation problems if the software was last updated in 2005 (sProUT). When we find the review, which is published in 2019 (e.g., [7]), the reader may assume that all analysed tools are available without limitations. In fact, many tools for ontology learning are not being updated for a long time or not at all. These conclusions emerged from Table 7, where only FRED machine reader satisfies the majority of parameters.

If we look more at the research and the comparative study-based journal articles, three tools for ontology learning are directly used in these articles. Authors of the article [23] introduced the logical patterns, called Open Knowledge Extraction motifs-fundamental blocks, of the RDF graphs. These motifs are used for the partition of the RDF graphs into semantically similar subsets. Experience with FRED is practically demonstrated. This tool is used for induction of RDF graphs from unstructured texts and its outputs are used for motifs identification. Motifs can be practically used to:

> *"annotate the OKE graph banks, to build OKE benchmarks, to evaluate OKE tools, to compare heterogeneous tools, and to perform on-demand OKE graph transformations."*

The same solution is used in [30], the author's present so-called MERGILO project (experiment), where the main aim is to find directions how to use multiple textual sources for building a knowledge graph (RDF-based formal structure). This process contains series of steps, including the knowledge reconciliation method, which is based on frame semantics and network alignment. The next part of the process is based on the transformation of the texts, expressed in natural language, into the RDF/OWL ontologies. This is realized by FRED. The authors point out the limits of FRED, where some problems with semantics expression can occur. OWL restrictions or disjointedness cannot be identified by the FRED at present. The process of the RDF/OWL ontology induction is described in more details in [30].

Text2Onto is the third tool directly used to solve a specific problem and mentioned in the research journal article [46]. One of the experiences with this tool is already mentioned in Section 4.2.2. Two different research articles, which are not part of our review, mention experience with the Text2Onto. Text2Onto is qualitatively evaluated by IT and non-IT users (published in 2009) [112]. Various criteria, including intuitiveness of the ontology building, ease of interactions/manipulations with the tool, positive or negative aspects, or tool expectation are used for evaluation of the tool. In the case of the parameter tool expectation, almost equal opinion (yes, no) is true for both types of users. The partial expectation is true for 25.9% of both types of users. More details are found in [46]. Authors of the second research article present an evaluation study, where manually developed ontology and automatically built ontology in Text2Onto are mutually compared (published in 2013) [55].

Three directions are considered for future research. As it is obvious, according to the brief comparative analysis of tools for ontology learning or knowledge extraction, FRED machine reader fulfils a majority of predefined requirements. This solution provides a Python-based library, called fredlib, for building its own applications, which are able to build ontologies on the basis of unstructured texts automatically. This library is poorly documented at present, but the authors declare that this documentation is in progress. Building one's own solution using the fredlib library, where batch processing of documents is available, is taken into account for future research.

SpaCy is a free, open-source, and regularly updated programming library which is focused on building its own NLP-based systems in Python [113]. SpaCy provides a big collection of functionalities, including the most fundamental linguistics operations, such as parts of speech tagging, lemmatization, sentences segmentation, morphological analysis, furthermore named entity recognition, comparing words or documents, or classification of texts. SpaCy is very similar to the well-known Python library called NLTK (Natural Language Toolkit) [114] which is recommended as a library for educational and research purposes. In comparison to the NLTK, spaCy supports integrated word vectors, dependency parsing, entity linking, and neural network models. NLTK provides larger collection of algorithms where not all of them are the best of all. SpaCy currently contains the best algorithm for particular problem. A developer no longer needs to wonder which algorithm should be chosen for the problem. This is the second possibility, which is going to be deeply investigated for ontology learning.

KNIME is the open-source platform mainly used for processing large collections of data [115,116]. A visual programming paradigm is applied for building so-called data science workflows, consisting of a collection of nodes, which are the smallest programming units in the KNIME. Each node provides one specific functionality. Suitable nodes are connected into the thread, which then processes data. Nodes for texts processing are also supported; nodes for taxonomy induction or ontology learning are not directly provided yet. On the other hand, nodes for opening and querying the RDF files are available. KNIME supports the development's own nodes, which suit different requirements. In this point of view, usage of already existing nodes together with own node(s) for taxonomy induction would be beneficial.

Collection of unstructured texts is going to be semi-automatically downloaded by the Zotero—freely available references management tool [117]. These texts will help us to realize experiments with the three above-mentioned solutions, where one is based on the visual programming approach (KNIME). We suppose that combination of already existed KNIME nodes will not be sufficient for realisation the ontology learning-based activities, but the new KNIME nodes have to be developed. Which KNIME nodes are going to be developed? This is the question which has to be answered in the future steps. Next, two tools (fredlib, spaCy) are going to be used straightforwardly, i.e., for relevant modules development, where each one will be responsible for particular ontology learning task. Taxonomy induction will be the first step towards the ontology learning. Unstructured texts, focused on the disaster management, are going to be used for the above mentioned experiments.

6. Conclusions

This paper aims to answer the question of which tool, applicable to ontology learning or knowledge extraction, can be practically used at present with respect to the predefined requirements. Unstructured texts related to the disaster management are in the centre of the interest. During the exploration of various solutions, simplicity of tool´s usage (an important aspect for unacquainted users in ontological engineering or programming) is also considered.

The final list, containing eight tools, is based on the review of full-text journal articles (research, review, or comparative study papers) available in ScienceDirect, WOS, and tools presented in the ProgrammableWeb.com website (accessed on 10 October 2021). Tools cited in these sources are manually extracted and thoroughly studied in the next phase. Only tools where information about ontology learning, taxonomy induction, or knowledge extraction is mentioned are selected for the comparative analysis and testing. It is important to find out whether these tools can be practically used for the above-mentioned activities and how many requirements are placed on the potential users, especially in view of the quality of documentation, ease of the installation process, degree of automation of the unstructured texts, or intuitiveness of the tool.

The review points out that an insufficient number of usable, stable and updated tools for ontology learning exist. It is essential to mention that this review emphasizes what is generally known. Automatic extraction of knowledge from unstructured texts is a big challenge. There is a gap in the world of tools used for ontology learning/knowledge extraction. Only a limited number of tools can be practically used by the final users. Key users from the target group do not have a strong background in programming, natural language processing, ontological engineering (learning), or information extraction. Deeper exploration of the OLTs is based on the analysis of the unstructured text related to the tsunami (meteotsunami) research, where the main aim is to assist the disaster management researchers and interested persons in processing extensive collections of unstructured documents. It has to be said that tools mentioned in this paper can be applied not only in disaster management but also in various different domains, as intelligent processing of textual information or knowledge does not have the only priority in this domain.

The main aim of the paper is not to apply rigorous evaluation measures for comparison of the tools, but it is to share the experience with similarly interested researchers who would like to find a solution for the facilitation of mining the knowledge from unstructured texts.

Supplementary Materials: Supplementary Materials are available online at https://www.mdpi.com/article/10.3390/info13010004/s1.

Author Contributions: Conceptualization, M.H., D.P. and T.N.; methodology, M.H. and D.P.; software, M.H. and M.Z.; validation, F.B.; formal analysis, M.H., D.P. and K.M.; investigation, M.H.; resources, P.Č. and K.Š.; data curation, P.M. and P.T.; writing—original draft preparation, M.H., D.P. and I.T.; writing—review and editing, M.H. and D.P.; visualization, M.H.; supervision, V.B. and F.B.; project administration, V.B.; funding acquisition, V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the VES20 Inter-Cost LTC 20020 project.

Data Availability Statement: Review data and studies together with the identification of tested tools are available in the Supplementary Materials.

Acknowledgments: The VES20 Inter-Cost LTC 20020 project supported this research. The authors also express gratitude to the COST Action AGITHAR leaders and team members.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Miner, G.; Elder, J.; Nisbet, R. *Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications*; Academic Press: Cambridge, MA, USA, 2012. [CrossRef]
2. Husáková, M. *Ontology-Based Conceptualisation of Text Mining Practice Areas for Education*; Springer: Cham, Switzerland, 2019; pp. 533–542. [CrossRef]
3. Gómez-Pérez, A.; Manzano-Macho, D. An overview of methods and tools for ontol- ogy learning from texts. *Knowl. Eng. Rev.* **2004**, *19*, 187–212. [CrossRef]
4. Barforoush, A.A.; Rahnama, A. Ontology Learning: Revisited. *J. Web Eng.* **2012**, *11*, 269–289.

5. Gangemi, A. A Comparison of Knowledge Extraction Tools for the Semantic Web. In *The Semantic Web: Semantics and Big Data*; Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 351–366. [CrossRef]
6. Alarfaj, A.; Al-Salman, A. Ontology Construction from Text: Challenges and Trends. *Int. J. Artif. Intell. Expert Syst.* **2015**, *6*, 15–26.
7. Konys, A. Knowledge Repository of Ontology Learning Tools from Text. *Procedia Comput. Sci.* **2019**, *159*, 1614–1628. [CrossRef]
8. Paris, R.; Goto, K.; Goff, J.; Yanagisawa, H. Advances in the study of mega-tsunamis in the geological record. *Earth-Sci. Rev.* **2020**, *210*, 103381. [CrossRef]
9. Goff, J.; Terry, J.P.; Chagué-Goff, C.; Goto, K. What is a mega-tsunami? In the wake of the 2011 Tohoku-oki tsunami—three years on. *Marine Geology* **2014**, *358*, 12–17. [CrossRef]
10. Costa, P.J.; Dawson, S.; Ramalho, R.S.; Engel, M.; Dourado, F.; Bosnic, I.; Andrade, C. A review on onshore tsunami deposits along the Atlantic coasts. *Earth-Sci. Rev.* **2021**, *212*, 103441. [CrossRef]
11. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am.* **2001**, *284*, 34–43. [CrossRef]
12. Husáková, M.; Bureš, V. Formal Ontologies in Information Systems Development: A Systematic Review. *Information* **2020**, *11*, 66. [CrossRef]
13. Protégé A free, Open-Source Ontology Editor and Framework for Building Intelligent Systems. Available online: https://protege.stanford.edu/ (accessed on 29 September 2021).
14. Wątróbski, J. Ontology learning methods from text—An extensive knowledge-based approach. *Procedia Comput. Sci.* **2020**, *176*, 3356–3368. [CrossRef]
15. Asim, M.N.; Wasim, M.; Khan, M.U.G.; Mahmood, W.; Abbasi, H.M. A Survey of Ontology Learning Techniques and Applications. Database 2018. 2018. Available online: https://academic.oup.com/database/article-pdf/doi/10.1093/database/bay101/27329264/bay101.pdf (accessed on 10 October 2021).
16. Khadir, A.C.; Aliane, H.; Guessoum, A. Ontology learning: Grand tour and challenges. *Comput. Sci. Rev.* **2021**, *39*, 100339. [CrossRef]
17. Buitelaar, P.; Cimiano, P. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*; IOS Press: Amsterdam, The Netherlands, 2008; Volume 167.
18. Maynard, D.; Bontcheva, K.; Augenstein, I. *Natural Language Processing for the Semantic Web*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2016.
19. Gangemi, A.; Presutti, V.; Recupero, D.R.; Nuzzolese, A.G.; Draicchio, F.; Mongiovì, M. Semantic Web Machine Reading with FRED. *Semant. Web* **2017**, *8*, 873–893. [CrossRef]
20. De Maio, C.; Fenza, G.; Loia, V.; Orciuoli, F. Unfolding social content evolution along time and semantics. *Future Gener. Comput. Syst.* **2017**, *66*, 146–159. [CrossRef]
21. Rani, M.; Dhar, A.K.; Vyas, O.P. Semi-automatic terminology ontology learning based on topic modeling. *Eng. Appl. Artif. Intell.* **2017**, *63*, 108–125. [CrossRef]
22. Boufrida, A.; Boufaida, Z. Rule extraction from scientific texts: Evaluation in the specialty of gynecology. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *108*, 33–41. [CrossRef]
23. Gangemi, A.; Recupero, D.R.; Mongiovì, M.; Nuzzolese, A.G.; Presutti, V. Identifying motifs for evaluating open knowledge extraction on the Web. New Avenues in Knowledge Bases for Natural Language Processing. *Knowl. Based Syst.* **2016**, *108*, 33–41. [CrossRef]
24. Konys, A. Towards Knowledge Handling in Ontology-Based Information Extraction Systems. *Procedia Comput. Sci.* **2018**, *126*, 2208–2218. [CrossRef]
25. Piad-Morffis, A.; Gutiérrez, Y.; Muñoz, R. A corpus to support eHealth Knowledge Discovery technologies. *J. Biomed. Inform.* **2019**, *94*, 103172. [CrossRef]
26. Zouaq, A.; Gagnon, M.; Jean-Louis, L. An assessment of open relation extrac-tion systems for the semantic web. *Inf. Syst.* **2017**, *71*, 228–239. [CrossRef]
27. De Rosa, M.; Fenza, G.; Gallo, A.; Gallo, M.; Loia, V. Pharmacovigilance in the era of social media: Discovering adverse drug events cross-relating Twitter and PubMed. *Future Comput. Syst.* **2021**, *114*, 394–402. [CrossRef]
28. Liu, S.; Yang, H.; Li, J.; Kolmanic, S. Preliminary Study on the Knowledge Graph Construction of Chinese Ancient History and Culture. *Information* **2020**, *11*, 186. [CrossRef]
29. Konys, A.; Drążek, Z. Ontology Learning Approaches to Provide Domain-Specific Knowledge Base. *Procedia Comput. Sci.* **2020**, *176*, 3324–3334. [CrossRef]
30. Mongiovì, M.; Recupero, D.R.; Gangemi, A.; Presutti, V.; Consoli, S. Merging open knowledge extracted from text with MERGILO. New Avenues in Knowledge Bases for Natural Language Processing. *Knowl. Based Syst.* **2016**, *108*, 155–167. [CrossRef]
31. Remolona, M.F.M.; Conway, M.F.; Balasubramanian, S.; Fan, L.; Feng, Z.; Gu, T.; Kim, H.; Nirantar, P.M.; Panda, S.; Ranabothu, N.R.; et al. Hybridontology-learning materials engineering system for pharmaceutical products: Multi-label entity recognition and concept detection. In honor of Professor Rafiqul Gani. *Comput. Chem. Eng.* **2017**, *107*, 49–60. [CrossRef]
32. Wohlgenannt, G.; Sabou, M.; Hanika, F. Crowd-based ontology engineering with the uComp Protege plugin. *Semant. Web* **2016**, *7*, 379–398. [CrossRef]
33. Rupasingha, R.A.H.M.; Paik, I.; Kumara, B.T.G.S. Specificity-Aware Ontology Generation for Improving Web Service Clustering. *IEICE Trans. Inf. Syst.* **2018**, *E101D*, 2035–2043. [CrossRef]

34. Rijvordt, W.; Hogenboom, F.; Frasincar, F. Ontology-Driven News Classification with Aethalides. *J. Web Eng.* **2019**, *18*, 627–654. [CrossRef]
35. Mohan, M.J.; Sunitha, C.; Ganesh, A.; Jaya, A. A Study on Ontology Based Abstrac-tive Summarization. *Procedia Comput. Sci.* **2016**, *87*, 32–37. [CrossRef]
36. Amar, F.B.B.; Gargouri, B.; Hamadou, A.B. Generating core domain ontologies from normal-ized dictionaries. Mining the Humanities: Technologies and Applications. *Eng. Appl. Artif. Intell.* **2016**, *51*, 230–241. [CrossRef]
37. Demner-Fushman, D.; Rogers, W.J.; Aronson, A.R. MetaMap Lite: An evaluation of a new Java implementation of MetaMap. *J. Am. Med Inform. Assoc.* **2017**, *24*, 841–844. [CrossRef]
38. Mezghanni, I.B.; Gargouri, F. CrimAr: A Criminal Arabic Ontology for a Benchmark Based Evaluation. *Procedia Comput. Sci.* **2017**, *112*, 653–662. [CrossRef]
39. Hoxha, J.; Jiang, G.; Weng, C. Automated learning of domain taxonomies from text using background knowledge. *J. Biomed. Inform.* **2016**, *63*, 295–306. [CrossRef]
40. Roldan-Molina, G.R.; Ruano-Ordas, D.; Basto-Fernandes, V.; Mendez, J.R. An ontology knowledge inspection methodology for quality assessment and continuous improvement. *Data Knowl. Eng.* **2021**, *133*, 101889. [CrossRef]
41. Barki, C.; Rahmouni, H.B.; Labidi, S. Model-based prediction of oncotherapy risks and side effects in bladder cancer. *Procedia Comput. Sci.* **2021**, *181*, 818–826. [CrossRef]
42. Ghoniem, R.M.; Alhelwa, N.; Shaalan, K. A Novel Hybrid Genetic-Whale Optimiza-tion Model for Ontology Learning from Arabic Text. *Algorithms* **2019**, *12*, 182. [CrossRef]
43. Kethavarapu, U.P.K.; Saraswathi, S. Concept Based Dynamic Ontology Creation for Job Recommendation System. *Procedia Comput. Sci.* **2016**, *85*, 915–921. [CrossRef]
44. Potoniec, J. Mining Cardinality Restrictions in OWL. *Found. Comput. Decis. Sci.* **2020**, *45*, 195–216. [CrossRef]
45. Salatino, A.A.; Thanapalasingam, T.; Mannocci, A.; Birukou, A.; Osborne, F.; Motta, E. The Computer Science Ontology: A Comprehensive Automatically-Generated Taxonomy of Research Areas. *Data Intell.* **2020**, *2*, 379–416. [CrossRef]
46. Xu, D.; Karray, M.H.; Archimède, B. A knowledge base with modularized ontologies for eco-labeling: Application for laundry detergents. *Comput. Ind.* **2018**, *98*, 118–133. [CrossRef]
47. Levin, B.W.; Nosov, M.A. General Information on Tsunami Waves, Seaquakes, and Other Catastrophic Phenomena in the Ocean. In *Physics of Tsunamis*; Springer International Publishing: Cham, Switzerland, 2016; pp. 1–34. [CrossRef]
48. Papadopoulos, G.A.; Lorito, S.; Løvholt, F.; Rudloff, A.; Schindelé, F. Understanding Disaster Risk: Hazard Related Risk Issues, Section I: Geophysical risk.; Publications Office of the European Union. 2017. Available online: https://publications.jrc.ec.europa.eu/repository/handle/JRC102482 (accessed on 18 October 2021).
49. Salaree, A.; Mansouri, R.; Okal, E.A. The intriguing tsunami of 19 March 2017 at Bandar Dayyer, Iran: Field survey and simulations. *Nat. Hazards* **2018**, *90*, 1277–1307. [CrossRef]
50. Maedche, A. The TEXT-TO-ONTO Environment. In *Ontology Learning for the Semantic Web*; Springer: Boston, MA, USA, 2002; pp. 151–170. [CrossRef]
51. Raimond, Y.; Schreiber, G. RDF 1.1 Primer. W3C note, W3C. 2014. Available online: https://www.w3.org/TR/2014/NOrdf11-primer-20140624/ (accessed on 10 October 2021).
52. TopQuadrant. TopBraid: Powerful Integrated Development Environment. Available online: https://www.topquadrant.com/products/topbraid-composer/ (accessed on 5 October 2021).
53. Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K. *Introduction to WordNet: An On-line Lexical Database**; Oxford University: Oxford, UK, 1991; p. 3. [CrossRef]
54. Cimiano, P.; Völker, J. Text2Onto: A Framework for Ontology Learning and Data-Driven Change Discovery. In Proceedings of the 10th International Conference on Natural Language and Information Systems, Alicante, Spain, 15–17 June 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 227–238. [CrossRef]
55. Mittal, S. Tools for Ontology Building from Texts: Analysis and Improvement of the Results of Text2Onto. *IOSR J. Comput. Eng.* **2013**, *11*, 101–117. [CrossRef]
56. Guha, R.; Brickley, D. RDF Schema 1.1. W3C Recommendation, W3C. 2014. Available online: https://www.w3.org/TR/2rdf-schema-20140225/ (accessed on 10 October 2021).
57. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation, W3C. 2012. Available online: https://www.w3.org/TR/2012/REC-owl2-overview-20121211/ (accessed on 5 October 2021).
58. Kifer, M. Rules and Ontologies in F-Logic. In *Reasoning Web: First International Summer School 2005*; Eisinger, N., Małuszyn´ski, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 22–34. [CrossRef]
59. Ledvinka, M.; Křemen, P. Formalizing Object-Ontological Mapping Using F-Logic. In Proceedings of the International Joint Conference on Rules and Reasoning, Bolzano, Italy, 16–19 September 2019; Fodor, P., Montali, M., Calvanese, D., Roman, D., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 97–112.
60. AIFB. text2onto. Available online: https://code.google.com/archive/p/text2onto/downloads (accessed on 6 October 2021).
61. Harth, A. NeOn Homepage. Available online: http://neon-toolkit.org/wiki/Main_Page.html (accessed on 7 October 2021).
62. The University of Sheffield. GATE—General Architecture for Text Engineering. Available online: https://gate.ac.uk/ (accessed on 7 October 2021).

63. University, P. WordNet—A Lexical Database for English. Available online: https://wordnet.princeton.edu/ (accessed on 7 October 2021).
64. Problem Installing Last Text2onto Standalone Version. Available online: https://github.com/martysteer/text2onto/issues/1 (accessed on 6 October 2021).
65. Things to Remember while Installing Text2Onto. Available online: https://ryadyo.wordpress.com/2012/02/16/things-to-remember-while-installing-text2onto/ (accessed on 6 October 2021).
66. STLab. FRED—Machine Reading for the Semantic Web. Available online: http://wit.istc.cnr.it/stlab-tools/fred/#About (accessed on 4 October 2021).
67. Etzioni, O.; Banko, M.; Cafarella, M.J. *Machine Reading*; AAAI Press: Palo Alto, CA, USA, 2006; Volume 2, pp. 1517–1519.
68. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; et al. DBpedia—A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semant. Web J.* **2015**, *6*, 167–195. [CrossRef]
69. Wang, J.; Aryani, A.; Wyborn, L.; Evans, B. Providing Research Graph Data in JSON-LD Using Schema.org. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 1213–1218. [CrossRef]
70. Bekki, D. Combinatory Categorial Grammar as a Substructural Logic. In *New Frontiers in Artificial Intelligence*; Onada, T., Bekki, D., McCready, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 16–29.
71. Presutti, V.; Draicchio, F.; Gangemi, A. Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames. In *Knowledge Engineering and Knowledge Management*; ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 114–129.
72. Kamp, H.; van Genabith, J.; Reyle, U. Discourse Representation Theory. In *Handbook of Philosophical Logic*; Gabbay, D.M., Guenthner, F., Eds.; Springer: Dordrecht, The Netherlands, 2011; Volume 15, pp. 125–394. [CrossRef]
73. Tan, H.; Kaliyaperumal, R.; Benis, N. Ontology-Driven Construction of Domain Corpus with Frame Semantics Annotations. In *Computational Linguistics and Intelligent Text Processing*; Gelbukh, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 54–65.
74. Liu, W.; Wang, T.; Yang, Z.; Cao, J. A Context-Aware Computing Method of Sentence Similarity Based on Frame Semantics. In *Advanced Data Mining and Applications*; Yang, X., Wang, C.D., Islam, M.S., Zhang, Z., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 114–126.
75. STLab. STLAB—FRED. Available online: http://wit.istc.cnr.it/stlab-tools/fred/demo/? (accessed on 4 October 2021).
76. Morita, T.; Fukuta, N.; Izumi, N.; Yamaguchi, T. DODDLE-OWL: A Domain Ontology Construction Tool with OWL. In Proceedings of the Semantic Web—ASWC 2006, First Asian Semantic Web Conference, Beijing, China, 3–7 September 2017; Mizoguchi, R., Shi, Z., Giunchiglia, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 537–551.
77. Sikos, L.F. Description Logics: Formal Foundation for Web Ontology Engineering. In *Description Logics in Multimedia Reasoning*; Springer International Publishing: Cham, Switzerland, 2017; pp. 67–120. [CrossRef]
78. Baader, F.; Horrocks, I.; Lutz, C.; Sattler, U. *An Introduction to Description Logic*, 1st ed.; Cambridge University Press: Cambridge, MA, USA, 2017.
79. Kurematsu, M.; Iwade, T.; Nakaya, N.; Yamaguchi, T. *DODDLE II: A Domain Ontology Development Environment Using a MRD and Text Corpus*; IEICE Transactions: Tokyo, Japan, 2004; Volume 87, pp. 908–916.
80. Yamaguchi, T. Constructing Domain Ontologies Based on Concept Drift Analysis. In Proceedings of the IJCAI-99, Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden, 2 August 1999.
81. Yokoi, T. The EDR Electronic Dictionary. Commun. *ACM* **1995**, *38*, 42–44. [CrossRef]
82. Morita, T. DODDLE-OWL Documentation. Available online: http://docs.doddle-owl.org/en/latest/index.html (accessed on 5 October 2021).
83. Nakagawa, H.; Mori, T. A Simple but Powerful Automatic Term Extraction Method. In Proceedings of the COLING-02: COMPUTERM 2002: Second International Workshop on Computational Terminology, Taipei, Taiwan, 31 August 2002.
84. Tsuruoka, Y.; Tsujii, J. Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 467–474. [CrossRef]
85. Navigli, R.; Velardi, P.; Cucchiarelli, A.; Neri, F. Quantitative and Qualitative Evaluation of the OntoLearn Ontology Learning System. In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, 23–27 August 2004; p. 1043-es. [CrossRef]
86. Navigli, R.; Velardi, P. Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. Comput. *Linguist* **2004**, *30*, 151–179. [CrossRef]
87. Team, T.O. Welcome to Ontolearn's documentation! Available online: https://ontolearn-docs-dice-group.netlify.app/index.html (accessed on 11 October 2021).
88. UIMA, O.T.C. Unstructured Information Management Architecture (UIMA) Version 1.0. Available online: https://docs.oasis-open.org/uima/v1.0/uima-v1.0.html (accessed on 21 October 2021).
89. Kluegl, P.; Toepfer, M.; Beck, P.D.; Fette, G.; Puppe, F. UIMA Ruta Workbench: Rule-based Text Annotation. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, Dublin, Ireland, 23–29 August 2014; pp. 29–33.
90. Foundation, T.A.S. UIMA Overview SDK Setup, Written and Maintained by the Apache UIMA™ Development Community, Version 3.2. Available online: https://uima.apache.org/d/uimaj-current/overview_and_setup.pdf (accessed on 21 October 2021).

91. Gede, P.I.B.; Kumara, I.N.S.; Sudarma, M. Systematic Review of Text Mining Application Using Apache UIMA. *Int. J. Eng. Emerg. Technol.* **2020**, *5*, 42–51. [CrossRef]
92. OpenNLP Welcome to Apache OpenNLP. Available online: https://opennlp.apache.org/ (accessed on 21 October 2021).
93. Klie, J.C.; Bugert, M.; Boullosa, B.; de Castilho, R.E.; Gurevych, I. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, Santa Fe, NM, USA, 20–26 August 2018; pp. 5–9.
94. Julie Component Repository (JCoRe) 2.0. Available online: https://julielab.de/Resources/JCoRe.html (accessed on 21 October 2021).
95. Foundation, T.A.S. Welcome to the Apache UIMA project. Available online: https://uima.apache.org/ (accessed on 4 October 2021).
96. Toepfer, M.; Fette, G.; Beck, P.D.; Kluegl, P.; Puppe, F. Integrated Tools for Query-driven Development of Light-weight Ontologies and Information Extraction Components. In Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT, Dublin, Ireland, 23 August 2014; pp. 83–92. [CrossRef]
97. Cram, D.; Daille, B. Terminology Extraction with Term Variant Detection. In Proceedings of the ACL-2016 System Demonstrations, Berlin, Germany, 7–12 August 2016; pp. 13–18. [CrossRef]
98. Fiorelli, M.; Pazienza, M.T.; Stellato, A.; Turbati, A. CODA: Computer-Aided Ontology Development Architecture. *IBM J. Res. Dev.* **2014**, *58*, 14. [CrossRef]
99. Fiorelli, M.; Gambella, R.; Pazienza, M.T.; Stellato, A.; Turbati, A. Semi-Automatic Knowledge Acquisition through CODA. In *Modern Advances in Applied Intelligence*; Ali, M., Pan, J.S., Chen, S.M., Horng, M.F., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 78–87.
100. ART Research Group University of Rome, T.V. CODA. Available online: http://art.uniroma2.it/coda/team/ (accessed on 21 October 2021).
101. Becker, M.; Drozdzynski, W.; Krieger, H.U.; Piskorski, J.; Schäfer, U.; Xu, F. SProUT—Shallow Processing with Typed Feature Structures and Unification. In Proceedings of the International Conference on NLP (ICON 2002), Mumbai, India, 19–21 December 2002.
102. DFKI. What is SProUT? Available online: https://sprout.dfki.de/ (accessed on 21 October 2021).
103. Drozdzynski, W.; Krieger, H.U.; Piskorski, J.; Schäfer, U.; Xu, F. Shallow Processing with Unification and Typed Feature Structures—Foundations and Applications. *Künstliche Intell.* **2004**, *18*, 17–23.
104. Speck, R.; Ngonga Ngomo, A.C. Named Entity Recognition Using FOX. In Proceedings of the International Semantic Web Conference (Posters & Demos), Riva del Garda, Italy, 21 October 2014.
105. FOX Federated Knowledge Extraction Framework. Available online: https://fox.demos.dice-research.org/#!/home (accessed on 5 November 2021).
106. Usbeck, R.; Ngonga Ngomo, A.C.; Röder, M.; Gerber, D.; Coelho, S.; Auer, S.; Both, A. AGDISTIS—Graph-Based Disambiguation of Named Entities Using Linked Data. In *The Semantic Web—ISWC 2014*; Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; Volume 8796, pp. 457–471. [CrossRef]
107. Finkel, J.R.; Grenager, T.; Manning, C. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005; pp. 363–370. [CrossRef]
108. Ratinov, L.; Roth, D. Design Challenges and Misconceptions in Named Entity Recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL '09, Boulder, CO, USA, 4–5 June 2009; pp. 147–155.
109. Nadeau, D. *Balie—Baseline Information Extraction: Multilingual Information Extraction from Text with Machine Learning and Natural Language Techniques*; School Infornatics and Technological Eengineering University: Ottawa, ON, Canada, 2005.
110. Konys, A. Knowledge systematization for ontology learning methods. *Procedia Comput. Sci.* **2018**, *126*, 2194–2207. [CrossRef]
111. Zhong, R.Y.; Newman, S.T.; Huang, G.Q.; Lan, S. Big Data for supply chain management in the service and manufacturing sectors: Challenges, opportunities, and future perspectives. *Computers Industrial Engineering* **2016**, *101*, 572–591. [CrossRef]
112. Hatala, M.; Gasevic, D.; Siadaty, M.; Jovanovic, J.; Torniai, C. Utility of Ontology Extraction Tools in the Hands of Educators. In Proceedings of the 2009 IEEE International Conference on Semantic Computing, Berkeley, CA, USA, 14–16 September 2009; pp. 408–413. [CrossRef]
113. Explosion. spaCy 101: Everything You Need to Know. Available online: https://spacy.io/usage/spacy-101 (accessed on 12 October 2021).
114. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*; O'Reilly: Beijing, China, 2009; Available online: http://my.safaribooksonline.com/9780596516499 (accessed on 10 October 2021).
115. Meinl, T.; Jagla, B.; Berthold, M.R. 6—Integrated data analysis with KNIME. In *Open Source Software in Life Science Research*; Harland, L., Forster, M., Eds.; Woodhead Publishing Series in Biomedicine; Woodhead Publishing: Amsterdam, The Netherlands, 2012; pp. 151–171. [CrossRef]
116. Radosevic, N.; Duckham, M.; Liu, G.J.; Sun, Q. Solar radiation modeling with KNIME and Solar Analyst: Increasing environmental model reproducibility using scientific workflows. *Environ. Model. Softw.* **2020**, *132*, 104780. [CrossRef]
117. Zotero: Your Personal Research Assistant. Available online: https://www.zotero.org/ (accessed on 29 January 2021).