Ontology Learning from Text

Departing the Ontology Layer Cake

Abel Browarnik and Oded Maimon
Department of Industrial Engineering
Tel Aviv University
Tel Aviv, Israel
email: {abel, maimon}@eng.tau.ac.il

Abstract— **We analyze the ontology learning objectives, reviewing the type of input one would expect to meet when learning ontologies - peer-reviewed scientific papers in English, papers that undergo quality control. We analyze the Ontology Learning Layer Cake model and its shortcomings, proposing alternative models for ontology learning based on linguistic knowledge and existing, wide coverage syntactical, lexical and semantic resources, using constructs such as clauses. We conclude, after showing that the Ontology Learning Layer Cake has a low maximum F measure (probably below 0.6), that the alternatives should be explored.**

Keywords - Ontology Learning from text; Ontology Learning Layer Cake Model; Language Modeling; Clauses; Subsentences.

I. INTRODUCTION

We define an *ontology* as a formal, explicit specification of a shared conceptualization. Most available ontologies are crafted and maintained with human intervention. Ontologies represent reality, and as such, require frequent updates, making them both costly and difficult to maintain. Ontology Learning (OL) has been developed in order to overcome this problem. Learning is interpreted in the literature as the process of creating the ontology and populating it. In this paper, the goal of OL is defined to be the (at least semi) automatic extraction of knowledge, possibly structured as simple or composite statements, from a given corpus of textual documents, to form an ontology.

Most, if not all, OL approaches [11][12][14][22][24] follow a model named the Ontology Learning Layer Cake (OLC), and share many features, such as statistical based information retrieval, machine learning, and data and text mining, resorting to linguistics based techniques for certain tasks.

This paper will argue that the Ontology Learning Layer Cake approach is not the best choice for Ontology Learning. The paper reviews the following issues:

Understanding Ontology Learning from text. An ontology represents, in our view, a “portion” of the world that we are looking at, for example, toxicity of engineered nanoparticles (or nanotoxicity). Every new paper published on the subject may add a new entity or relationship to the nanotoxicity model. We see OL as a tool for modeling a domain and keeping this model updated.

The input used to learn ontologies. The input to the OL process depends on the domain itself. Modeling scientific domains, such as nanotoxicity, normally draws on peer-reviewed papers or other scientific articles, magazines or books. These are well-formed and quality-checked texts. We will argue that the quality of the input is one of the parameters to be taken into account when devising an OL framework.

Analysis of the Ontology Learning Layer Cake Model. The Ontology Learning Layer Cake model aims at learning ontologies by using a multistep approach. Most OLC driven methods rely on at some stage to statistics or machine learning, the basis for unsupervised learning methods, either to extract terms, to build taxonomies or to extract relations and rules. We will argue that the sequential nature of OLC results in rather low overall recall and precision for the whole process.

Alternative models for Ontology Learning. Based on the conclusions of our analysis of OL, the input to the OL process, and the OLC model, we find that the entire subject of OL could be tackled in a different manner. Assuming that many of the target domains are defined by well-formed texts, we introduce the fundamentals of alternative OL frameworks. These fundamentals build on English language structure.

The paper includes, following this Introduction, a background section, an review of alternative models and a discussion where statistical models are compared to linguistic methods. The paper also includes a conclusion section.

II. BACKGROUND

A. Understanding Ontology Learning from text

Ontology Learning from text aims to obtain knowledge on the domain covered by the text. It is often seen as the extraction of ontologies by applying natural language analysis techniques to texts.

Cimiano [9] describes the tasks involved in OL as forming a layer cake. The cake is composed, in ascending order, by terms, sometimes synonyms, concepts, taxonomies, relations and finally axioms and rules.

This approach can be seen as a cornerstone in OL. It assumes that terms (gathered through term extraction methods) are the basic building blocks for OL. There are many term extraction methods [3][19][29] and many tools are publicly available [1][24][28]. The synonym layer is either based on sets, such as WordNet synsets [23] (after sense

 ISBN: 978-1-61208-445-9

disambiguation), on clustering techniques [3][12][22][11] or other similar methods, or on web-based knowledge acquisition.

The concept layer perception depends on the definition of concept. The consensual view is that it should include: an intensional definition of the concept, a set of concept instances, i.e., its extension, and a set of linguistic realizations, i.e., (multilingual) terms for this concept.

The concept hierarchy level (i.e., the taxonomic level) uses one of three paradigms to induce taxonomies from text:

- The application of lexico-syntactic patterns to detect hyponymy relations [16]. This approach is known to have reasonable precision but very low recall.
- The exploitation of hierarchical clustering algorithms to automatically derive term hierarchies from text, based on Harris' distributional hypothesis [15], that terms are similar in meaning to the extent in which they share syntactic contexts.
- A document-based notion of term subsumption, as proposed, for example, in Sanderson and Croft [27]. Salient words and phrases extracted from the documents are organized hierarchically using a type of co-occurrence known as subsumption.

The relation level has been addressed primarily within the biomedical field. The goal is to discover new relationships between known concepts (such as symptoms, drugs, and diseases) by analyzing large quantities of biomedical scientific articles. Relation extraction through text mining for ontology development was introduced in work on association rules in Maedche and Staab [21]. Recent efforts in relation extraction from text have been carried on under the Automatic Content Extraction (ACE) program, where entities (i.e., individuals) are distinguished from their mentions. Normalization, the process of establishing links between mentions in a document and individual entities represented in a ontology, is part of the task for certain kind of mentions (e.g., temporal expressions).

The rule level is at an early stage [20]. The European Union-funded project Pascal [10] on textual entailment challenge has drawn attention to this problem.

Our analysis of OL takes Wong [30] and Wong, Liu and Bennamoun [31] as its starting point.

The following remarks represent the consensus among OL reviews:

- The fully automatic learning of ontologies may not be possible.
- A common evaluation platform for ontologies is needed.
- The results for discovery of relations between concepts are less than satisfactory.
- The more recent literature points to an increase in interest in using the Web to address the knowledge acquisition bottleneck and to make OL operational on a Web scale.

Ontology Learning starts at a given point in time. It collects the existing knowledge by using the methods available and builds a representation of this knowledge. There are many schemes for knowledge representation, such as Extensible Markup Language (XML), Resource Description Framework (RDF)/RDF Schema (RDFS), Web Ontology Language (OWL)/OWL2 and Entity-Relationship Diagrams (ERD).

The representation scheme chosen affects the extent of reasoning that the Ontology will allow. An XML represented Ontology will allow less reasoning than a First Order Logic scheme.

As knowledge is added, the representation absorbs the new knowledge incrementally. The scheme should not permit contradictory knowledge. Therefore, if new knowledge contradicts existing knowledge, a protocol is needed to resolve the contradictions.

New knowledge is created by scientific work published (e.g., books, papers, proceedings). The input is processed and incorporated into the knowledge representation.

B. *The Input Used to Learn Ontologies*

There are a few types of ontologies. Upper or foundation ontologies are general purpose ontologies and define reality. Domain ontologies, on the other hand, are used to depict a domain. A domain ontology plays a role similar to that of the conceptual layer of an ERD in the area of system analysis. In both cases the relevant concepts are entities, attributes, relationships and more.

System analysis is performed by humans – system analysts – that gather information from humans involved in the domain, together with environmental details, to create the conceptual layer of an ERD for that domain. An ERD has two additional layers, the logical layer and the physical layer. These two layers deal with implementation details and therefore are not relevant to our discussion. From now on, when we refer to ERD we mean the conceptual layer of an ERD. We argue that an ERD is equivalent to an ontology, because an ERD of the domain represents conceptually the entities involved and the relations between the entities. OL is the task of gathering the information necessary to build the ontology of the domain (and perhaps to populate it). This is similar to building an ERD, even though the means to build an ERD are not necessarily the same means required to learn an ontology.

The main difference between creating an ERD for a business and learning an Ontology for a domain is the fact that the domain builds on a body of scientific books or papers that are a strong basis for a learning process without human intervention (except for paper writing), while building an ontology for an information domain such as an Enterprise Requirements Planning (ERP) system relies on knowledge that is seldom written, let aside formalized. Yet, in both cases we target a model of the domain. Thus, we see OL as a modeling technique.

We should consider the sources of text used towards learning ontologies, and the quality of these texts. To this end we could think of a Martian visiting Earth. The visitor could find him/herself browsing the New York Times website on November 21st, 2013. He/She could see there that "Applicants Find Health Website Is Improving, but Not Fast Enough ". Having no worldly knowledge he/she would not understand that this issue is related to the United States (US)

 ISBN: 978-1-61208-445-9

health reform commonly referred to as Obamacare. This is where an ontology comes of use. An ontology of US politics would provide the visitor with the background knowledge he would need to understand the newspaper. The source for this OL task would be newspapers and books. As we deal with learning ontologies from text we do not consider video or audio sources. We do not consider new media, such as Tweeter, email or Facebook either, because language quality in new media cannot be taken for granted. Newspapers, magazines and books undergo editing which is a sort of quality control. This is not to say that there is no use for new media. It can be used for less formal tasks, as is the case with sentiment analysis.

Most existing methods for OL from text rely on well-formed text. There is no clear guidance on this issue. Our Literature review reveals that existing tools such as ASIUM [12], OntoLearn [24] and CRCTOL [17] perform term extraction using sentence parsing. Text-to-Onto [22], TextStorm/Clouds [25] and Syndikate [14] perform term extraction using syntactic structure analysis and other techniques. OntoGain [11] uses shallow parsing for term extraction. If the text is not well-formed, these tasks would not be feasible. Thus, we assume that the input for OL from text consists of well-formed text.

C. Analysis of the Ontology Learning Layer Cake Model

Methods using the Ontology Learning Layer Cake model divide the OL task into four or five sequential steps. These steps result in the following outputs:

- Terms
- Concepts
- Taxonomic relations
- Non-taxonomic relations
- Axioms

Some methods perform all the steps, while some perform only part of them. Recall and precision obtained by the methods vary. ASIUM [12], Text-to-Onto [22], Ontolearn [24] and Ontogain [11] do not provide an overall figure of precision and recall for the whole OL process. TextStorm/Clouds [25] cites an average result of 52%. Syndikate [14] mentions high precision (94 to 97 % for different domains) and low recall (31 to 57% correspondingly, for the same domains). CRCTOL [17] reports a figure of 90.3% for simple sentences and 68.6% for complex sentences (we assume that these figures represent the F measure of the method). The main characteristics shared by the methods based on the OLC model are:

- The method is split into sequential steps. The output of step i is the input for step $i + 1$ (though there may be additional inputs from other sources).
- Individual steps may produce, in addition to the main output expected, other results. As an example, ASIUM, OntoLearn and CRCTOL perform term extraction using sentence parsing. This can be considered a secondary output. Secondary output from step i is not passed to step $i + 1$.
- If a method has four or five sequential steps, each step depends on the previous one. If every step has precision and recall (and therefore their harmonic mean, the F measure) bound by p ($p < 1$), then the method cannot obtain recall and precision better than p^n (n is the number of steps). As an example, if we assume the F measure of each step to be 0.8, the F measure of the whole OL method with 4 steps will be 0.41. With 0.9 (a result seldom attained) per step the F measure is 0.59!
- A step which uses statistical or machine learning methods requires considerable amounts of data to give significant results. In general, it also requires the data to be split into a training set and a test set.
- Statistical and machine learning methods have to beware of the danger of over-fitting and wrong choices of training and test sets. These may result in output distortion.
- OLC methods require statistical evidence regarding knowledge of the area being studied. Thus, features such as co-occurrence of terms or words may induce conclusions that are nonsensical to subject experts.
- The statistical nature of some steps makes it impossible to trace back specific results. As an example, a method may find a relation between two concepts following the co-occurrence of the two concepts in the same sentence or paragraph in different portions of text, or even in different documents.

Often the unsupervised nature of statistical or machine learning methods is an incentive to choose such methods, as less human effort is required to understand the subject matter. Such understanding is critical for the success of non-statistical, non-machine learning methods. The human effort and the fact that results are sometimes similar for both supervised and unsupervised methods tip the scales, leading the practitioner to choose unsupervised methods. In this case, however, we see that OLC methods sometimes use supervised techniques. Such may be the case, for example, in TextStorm/Clouds. This method uses part of speech tagging (using WordNet), syntactic structure analysis, and anaphora resolution for any of the steps of the OLC process, for example, term extraction, and taxonomic and non-taxonomic relation learning. Yet, this is an OLC method with its “cascading” nature.

It is possible that the OLC approach was inspired by the “divide and conquer” algorithm design paradigm. A divide and conquer algorithm works by recursively breaking down a problem into smaller sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem. Problems in data mining are often solved using “cascading” algorithms built on the divide and conquer paradigm. The fact that data mining was followed by textual data mining which, in turn, inspired OL may be one of the reasons for choosing OLC.

III. Alternative Models for Ontology Learning

The approaches and methods reviewed above stem mainly from Cimiano’s ontology layer cake. That is, there is

 ISBN: 978-1-61208-445-9

consensus that first one has to gather terms (and probably also synonyms), then concepts, and finally extract relations (taxonomic for all the systems, with some of the systems and approaches aiming also at non-taxonomic relations, with a variable degree of success). In addition, few systems cross the reasoning threshold. Some of the methods are purely statistic; most use a mixture of statistical based and linguistic and/or Natural Language Processing (NLP) based methods, with statistics based methods taking an important role. The reason for this may be based on Brants conjecture [4]. Brants argues that NLP contribution to Information Retrieval related tasks is rather ineffective. Is this the only way to proceed? We would initially ask two questions:

- Would it be possible to start, for example, by gathering relations (any relation, not necessarily taxonomic) and then proceed to the other layers mentioned in the OLC?
- If we want to store knowledge in RDF or RDFS is there any requirement that the order should respect the OLC order?

It is worth mentioning that even linguistic or NLP based methods may rely on corpora. It is said that the most promising trend exploring the web as the corpus of choice due to its extent and coverage.

There is a third question:

- We are dealing with a specific and bound subject – ontology learning. Would it be appropriate to deal with text in a purely linguistic, even linguistic-theoretic manner? In other words, do we have to rely on corpora, or can we use language modeling to obtain results?

A. *Extracting semantic representation from text*

Research on Psycholinguistics and Neurolinguistics looks at how humans gather information from text (see for example, Caplan, [6]). It is generally agreed that humans gather information from text at the sentence level or even at the clause level, and not at the document (or corpus) level. Thus, extracting the semantic payload of text would ideally include deep parsing, semantic labeling of the text and a process of knowledge accumulation. From a practical point of view, the above may not be feasible. To overcome these limitations, researchers apply practical approaches based on heuristics and partial methods.

The literature shows several attempts to gather information from text at the sentence level. The model proposed by Chen and Wu [8] makes extensive use of FrameNet [13]. A semantic frame can be thought of as a concept with a script. It is used to describe an object, state or event. Chen and Wu avoid the need to deep-parse the sentences that constitute the text by using Framenet.

Chaumartin [7] presents another attempt to tackle the semantic representation issues. Instead of using Framenet, Antelope, the implementation of Chaumartin's work uses VerbNet [18] for the lexical-semantic transition.

Both methods (Chaumartin [7] and Chen and Wu [8]) deal with text at the sentence level, without taking into account sub-sentence components. Chen does not provide a tool to showcase the capabilities of his approach, except for an example in the paper: "*They had to journey from Heathrow to Edinburgh by overnight coach*. ". The example is assigned Framenet's frame *Travel* with all its elements (traveler, source and goal). Chaumartin released a full-fledged toolbox to test the capabilities of his approach. The system includes an example with its result, a clear semantic representation of the sentence in terms of VerbNet classes and all the resulting constraints. The representation includes all the semantic details necessary to assess the situation and allow for higher order activities such as question answering, reasoning and maybe automatic translation. Yet, for other sentences, results are not satisfactory, as in:

"Most of these therapeutic agents require intracellular uptake for their therapeutic effect because their site of action is within the cell"

The example above yields no result (i.e., no VerbNet class is recognized and therefore no semantic representation is extracted). One of the reasons for failing to discover the semantic contents of complex or compound sentences may be that such a sentence structure requires more than one frame or verb class to be found.

B. *From clauses or subsentences to RDF triples and RDFS*

The Resource Description Framework (RDF) data model, defined in http://www.w3.org/RDF/, makes statements about resources in the form of subject-predicate-object expressions known as triples. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object.

The Longman Grammar of Spoken and Written English (LGSWE) [2] defines a clause as a unit structured around a verb phrase. The lexical verb in the verb phrase denotes an action (drive, run, shout, etc.) or a state (know, seem, resemble, etc.). The verb phrase appears with one or more elements denoting the participants involved in the action, state, etc. (agent, affected, recipient, etc.), the attendant circumstances (time, place, manner, etc.), the relationship of the clause to the surrounding structures, etc. Together with the verb phrase, these are the clause elements. The clause elements are realized by phrases or by embedded clauses. A clause may be divided into two main parts: a subject and a predicate. The subject is generally a nominal part, while the predicate is mainly a verbal nucleus. Preisler [26] states that a clause contains the constituents Subject, Verbal, Complement and Adverbial, all or some of them. Rank shifting adds complexity to the subject. In this context, rankshifted clauses are called subclauses, while non-rankshifted clauses are called main clauses.

Clauses appearing together in a larger unit (generally sentences, but possibly phrases in the event of a rank-shifted clause) are linked by structural links, the principal types being coordinators, subordinators and wh-words. Coordinators create coordinated clauses. On the other hand, subordinators and wh-words create embedded clauses.

Subsentences, a concept introduced by Browarnik and Maimon [5] sometimes overlap with clauses. Yet, subsentences keep the construct simpler because of the restriction to the number of Verbal Constructs (VC) per subsentence.

The above definitions give a clue on to how to represent knowledge extracted by linguistic modeling by using RDF constructs, e.g., an RDF triple and a clause or a subsentence seem to represent entities and relationships. In other words, knowledge extracted from a clause or a subsentence can be represented by an RDF triple. Generally, an RDF triple is defined by an RDF scheme. In our case, as we start from knowledge extracted from a clause or a subsentence to obtain an RDF triple, the RDF scheme (or RDFS) should be obtained from a generalization of the RDF triples obtained, in a bottom-up fashion.

C. *Advantages of language modeling approaches*

Traceability. No matter whether one picks deep parsing or one of the heuristic methods (using sentences, clauses or subsentences), the result is that one sentence is derived into a set of RDF triples. It is possible that one RDF triple is derived from more than one sentence. This mechanism creates a clear relationship between the input sentence and the output RDF triple. A human reviewer may decide to check the Ontology Learning results. Such a review is feasible. Moreover, if the result turns out to be erroneous, it may be fixed. Such fixing may have an impact on the method and change other results.

Contradictory facts. If two scientific papers contain contradictory statements and both statements are used towards learning an Ontology we face a problem. While it is possible that the issue remains unresolved in the scientific community, we cannot assert both results in the resulting Knowledge Base. This situation is better than what a Machine Learning approach would provide, i.e., a statistical procedure that would add the most statistically significant result into the Knowledge Base, making it hard to clarify afterwards whether the result was appropriate.

Big Corpora. Most OLC methods are based on statistical processing. The corpus has to be split into a training set and a test set. The outcome is measured by recall and precision. As mentioned before, the performance of each step in an OLC method is bounded and therefore the result of a 4-step cascading method is theoretically bounded. On the other hand, Language Modeling does not require the use of big corpora, at least explicitly.

Recall and Precision. Recall measures the percentage of results that should have been returned by the method used. Methods based on Language Modeling make this measure less relevant. The methods process every sentence on the input text and return a result. Therefore one could argue that recall would always be 100%. Precision, while still relevant to the Language Modeling methods, may be interpreted differently. The methods do return a result, yet the result may be wrong, therefore reducing precision. If and when the mistake is discovered, the traceability mentioned above can be used to correct it, thus improving the model's precision.

IV. DISCUSSION: STATISTICAL VS. LINGUISTIC-BASED METHODS

The attempt to construct a model follows one of two possible approaches, and sometimes a mixture of the two approaches.

The Language Modeling approach aims at understanding the subject matter. Such approaches generally rely on a thorough knowledge of the subject matter. The results are generally accurate. When results accumulate they either confirm the adequacy of the model, making it widely accepted, or undermine it, leading it ultimately to be discarded. Physics shows plenty of theoretical models that were accepted after obtaining more and more experimental confirmations. Even more models were rejected after experimental evidence showed they were wrong.

The other approach aims at creating models by gathering facts and statistics that give us a hint about the "internals" of the subject matter, without obtaining a detailed understanding of these internals. Engineering and Medicine are areas where such methods flourish. A good example is the area of queuing theory. To forecast the arrivals of requests, one often uses heuristics to decide on a given probability distribution. Such decisions give a good approximation to the real conditions of the problem, but not necessarily the best theoretical fit.

Most methods for Natural Language Processing (NLP), and especially the methods used for OL, draw on the second approach. To mention only the most prominent OL systems, we see that:

- ASIUM [12] uses agglomerative clustering for taxonomy relations discovery.
- Text-To-Onto [22] uses agglomerative clustering, hypernyms from WordNet and lexico-syntactic patterns for taxonomic relation extraction. Non-taxonomic relations are extracted using association rule mining.
- In TextStorm/Clouds [25], both taxonomic and non-taxonomic relations are obtained using part of speech tagging, WordNet, syntactic structure analysis and anaphora resolution.
- Syndikate [14] implements semantic templates and domain knowledge for relations extraction.
- OntoLearn [24] relation extraction relies on hypernyms from WordNet (relations extracted are only taxonomic).
- CRCTOL [17] achieves relation extraction (taxonomic and non-taxonomic) using lexico-syntactic patterns and syntactic structure analysis.
- OntoGain [11] applies aglomerative clustering and formal concept analysis to extract taxonomic relations and Association rule mining for non-taxonomic relations.

Moreover, for most of the reviewed OL methods and systems, even the term and concept layers (stemming from Cimiano's ontology layer cake) are extracted using statistical methods.

The Language Modeling approaches for OL from English texts are based on the following facts:

- An ontology can be represented by RDF triples.
- RDF triples are subject-predicate-object expressions.
- Clauses are components of sentences and include a subject, a verbal part, a complement and an adverbial

part, all or some of them. Subsentences are a textual passage built around one verbal construct.

- RDF triples are equivalent to clauses or subsentences.

Therefore, an RDF triple can be constructed from a clause or a subsentence. But, how does one extract the triple from a clause? And how does one find a clause from a sentence? Our preferred solution is to use Clause Boundary Detection or Subsentence Detection. Both are characterized by near linear time complexity.

Chaumartin shows how to extract a kind of role based frame from a sentence, although, as we indicated above, working at the sentence level has succeeded only partially.

Yet, statistical methods are very useful and should by no means be neglected. Constructing resources, such, as part of speech taggers, WordNet, VerbNet or FrameNet, do profit from statistical methods. Based on these resources, one can create a somehow theoretical linguistic model that would not rely on corpora in order to extract clauses or subsentences from sentences, and in turn convert it into RDF triples, thus learning an ontology with no – direct – use of corpora.

V. CONCLUSION

We have shown that OLC methods have generally low recall and precision (less than 0.6). We have shown that OL is generally based on well-formed text, and that text can be decomposed into clauses (or subsentences), and then translated into RDF statements. The Language Modeling approaches make backtracking results possible, therefore allowing for a correction option. The approaches do not rely on big corpora, making these approaches potentially more efficient than OLC. The Language Modeling approaches elude OLC problems such as bounded performance due to OLC cascading nature, limitations of its statistical basis, the need for big corpora, and the problems associated with such corpora.

REFERENCES

[1] M. Baroni and S. Bernardini, (2004), “BootCaT: Bootstrapping corpora and terms from the web”, In Proceedings of LREC (vol. 4), L04-1306, Lisbon, Portugal.

[2] D. Biber, S. Johansson, G. Leech, S. Conrad, and E. Finegan, (1999), Longman Grammar of Spoken and Written English, Longman Publications Group, ISBN 0-582-23725-4.

[3] D. Bourigault and C. Jacquemin, (1999), “Term extraction+ term clustering: An integrated platform for computer-aided terminology”, In Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, (pp. 15-22), Association for Computational Linguistics.

[4] T. Brants, (2003), “Natural Language Processing in Information Retrieval”, In Bart Decadt, Véronique Hoste, Guy De Pauw (Eds.): Computational Linguistics in the Netherlands 2003, December 19, Centre for Dutch Language and Speech, University of Antwerp. Antwerp papers in Linguistics University of Antwerp 2003.

[5] A. Browarnik and O. Maimon, (2012), “Subsentence Detection with Regular Expressions”, Presented at the XXX AESLA International Conference, Lleida, April 2012.

[6] D. Caplan, (2003), “Neurolinguistics”, In The Handbook of Linguistics, UK: Blackwell Publishing.

[7] F. R. Chaumartin, (2005), “Conception and Implementation of a Syntactic/Semantic Interface Using English Large Coverage Resources”. Master Thesis in Linguistics and Information Technology (in French), Universite Paris7 – Denis Diderot, 2005.

[8] E. Chen and G. Wu, (2005), “An Ontology Learning Method Enhanced by Frame Semantics”, ISM 2005: 374-382.

[9] P. Cimiano, (2006), “Ontology Learning and Population from Text.Algorithms, Evaluation and Applications”, ISBN: 978-0-387-30632-2, Springer, 2006

[10] I. Dagan, O. Glickman, and B. Magnini, (2006), “The PASCAL Recognising Textual Entailment Challenge”. Lecture Notes in Computer Science, Volume 3944, Jan 2006, Pages 177 - 190.

[11] E. Drymonas, K. Zervanou, and E. Petrakis, (2010), “Unsupervised ontology acquisition from plain texts: The OntoGain system”, Natural Language Processing and Information Systems, 277-287.

[12] D. Faure and C. Nedellec, (1999), “Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM”. Knowledge Acquisition, Modeling and Management, 329-334.

[13] C. J. Fillmore, (1976), “Frame semantics and the nature of language”, Annals of the New York Academy of Sciences, (280):20–32.

[14] U. Hahn, M. Romacker, and S. Schulz, (2000), “MedSynDiKATe--design considerations for an ontology-based medical text understanding system”, In Proceedings of the AMIA Symposium (p. 330), American Medical Informatics Association.

[15] Z. Harris, (1968), Mathematical Structures of Language, John Wiley & Sons, 1968.

[16] M. A. Hearst, (1992), “Automatic acquisition of hyponyms from large text corpora”, In Proceedings of the 14th conference on Computational linguistics-Volume 2 (pp. 539-545), Association for Computational Linguistics.

[17] X. Jiang and A. H. Tan, (2009), “CRCTOL: A semantic‐based domain ontology learning system”, Journal of the American Society for Information Science and Technology, 61(1), 150-168.

[18] K. Kipper Schuler, (2005), “VerbNet - a broad-coverage, comprehensive verb lexicon”, PhD thesis, University of Pennsylvania. http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf

[19] L. Kozakov, Y. Park, T. Fin, Y. Drissi, Y. Doganata, and T. Cofino, (2004), “Glossary extraction and utilization in the information search and delivery system for IBM Technical Support”, IBM Systems Journal, 43(3), 546-563.

[20] D. Lin and P. Pantel, (2001), “DIRT - Discovery of Inference Rules from Text”, In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2001, pp. 323-328.

[21] A. Maedche and S. Staab, (2000), “Discovering conceptual relations from text”. In W. Horn, editor, Proceedings of the 14th European Conference on Artificial Intelligence (ECAI’2000), 2000.

[22] A. Maedche and S. Staab, (2000, August), “The text-to-onto ontology learning environment”, In Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures.

[23] G. A. Miller , R. Beckwith , C. Fellbaum , D. Gross , and K. J. Miller, "Introduction to WordNet: An On-line Lexical Database", Int J Lexicography 3: 235-244.

[24] R. Navigli and P. Velardi, (2004), “Learning Domain Ontologies from Document Warehouses and Dedicated Websites”, Computational Linguistics, 30(2), MIT Press, 2004, pp. 151-179.

[25] A. Oliveira, F. C. Pereira, and A. Cardoso, (2001), "Automatic reading and learning from text", In Proceedings of the International Symposium on Artificial Intelligence (ISAI).

[26] B. Preisler, (1997), A Handbook of English Grammar on Functional Principles, Aarhus University Press, Denmark, ISBN 87 7288 405 3.

[27] M. Sanderson and B. Croft, (1999), "Deriving concept hierarchies from text", In Research and Development in Information Retrieval, pages 206–213. 1999.

[28] F. Sclano and P. Velardi, (2007), "TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities". To appear in Proc. of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007), Funchal (Madeira Island), Portugal, March 28–30th, 2007.

[29] J. Wermter and U. Hahn, (2005), "Finding new terminology in very large corpora", In Proceedings of the 3rd international conference on Knowledge capture (pp. 137-144), ACM.

[30] W. Y. Wong, (2009), "Learning lightweight ontologies from text across different domains using the web as background knowledge", (Doctoral dissertation, University of Western Australia).

[31] W. Wong, W. Liu, and M. Bennamoun, (2012), "Ontology Learning from Text: A Look back and into the Future", ACM Computing Surveys (CSUR), 44(4), 20.