A Survey on Ontology Enrichment from Text

by

Vivek Iyer, LALIT Mohan Mohan, Y.Raghu Babu Reddy, Mehar Bhatia

in

The sixteenth International Conference on Natural Language Processing (ICON-2019)

Hyderabad, India

Report No: IIIT/TR/2019/-1

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2019

A Survey on Ontology Enrichment from Text

Vivek Iyer
IIIT, Gachibowli
Hyderabad - 500032

Lalit Mohan
IIIT, Gachibowli
Hyderabad - 500032

Y. Raghu Reddy
IIIT, Gachibowli
Hyderabad - 500032

Mehar Bhatia
Shiv Nadar University
Greater Noida
India - 201314

Abstract

Increased internet bandwidth at low cost is leading to the creation of large volumes of unstructured data. This data explosion opens up opportunities for the creation of a variety of data-driven intelligent systems, such as the Semantic Web. Ontologies form one of the most crucial layers of semantic web, and the extraction and enrichment of ontologies given this data explosion becomes an inevitable research problem. In this paper, we survey the literature on semi-automatic and automatic ontology extraction and enrichment and classify them into four broad categories based on the approach. Then, we proceed to narrow down four algorithms from each of these categories, implement and analytically compare them based on parameters like context relevance, efficiency and precision. Lastly, we propose a Long Short Term Memory Networks (LSTM) based deep learning approach to try and overcome the gaps identified in these approaches.

1 Introduction

There has been an explosion of data on the Internet in the past few years, primarily caused by the drastic increase in the number of internet users over the years. About 90% of the data on internet has been created since 2016, mainly because of the massive increase in the user base and machine to machine communication. Data is defined as unprocessed facts and figures that do not contain any added interpretation or analysis. Information is interpretation of structured or unstructured data so that it holds meaning. Knowledge is processed information, experience, and insight combined such that it is beneficial to the end user[1].

Web pages, the primary source of *knowledge* on the World Wide Web (WWW) are primarily text documents annotated using Hypertext Markup Language (HTML). Lack of semantic markup of pages can result in irrelevant search results. The semantic web[2] provides a format or structure to machines to understand the *meaning* of the web page data rather relying on HTML markup, to make web intelligent and intuitive to user's queries. The semantic web includes data-centric publishing languages, including RDF (Resource Description Framework - the data modeling language for the semantic web), SPARQL (SPARQL protocol and RDF query language for semantic web) and OWL (Web Ontology Language - schema language, or knowledge representation language, of the semantic web), which allows meaning and structure to be added to content in a machine-readable format. OWL [3] allows definition of concepts composably, i.e. in such a way that it allows the reuse of concepts and relationships. Given the amount of information being extracted from the data generated on a regular basis in various domains, it becomes essential for it to be stored in the form of knowledge in ontologies. However, the knowledge stored in ontologies is rarely static. Like all other knowledge structures, its vital for ontologies to be enriched with time so as to improve the quality of search results.

Given recent advances in the fields of artificial intelligence and machine learning, as well as increased data processing capabilities with increase in compute power, newer, better and more accurate ways of extracting and enriching ontologies from text are now possible. Ontology extraction from text has primarily been at lower layers in ontology ”layer cake” (Buitelaar et al.,

[1] https://tinyurl.com/datainfknowledge

[2] https://expertsystem.com/what-is-the-semantic-web/

[3] https://db-x.org/blog/2016/04/15/semantic-web-2/

2005). A pre-existing seed ontology created manually or through learning needs enrichment. Ontology enrichment (Faatz and Steinmetz, 2002) is population, updation, and adaptation (Noy and Klein, 2004) of concepts, relations and rules. In the context of this paper, we assume a pre-existing seed ontology that is enriched by learning (semi-automatic or automatic) from text. The survey in this paper attempts to address the following important research questions:

- How are ontologies enriched by learning from unstructured text, and which algorithms are considered seminal?
- How do seminal algorithms compare with each other, in regards to context relevance, algorithmic efficiency and precision?
- What gaps are identified in these algorithms, and how can they be potentially addressed?

We did not focus on the other knowledge representation methods such as knowledge graphs, frames, semantic nets and others in the survey considering the extensivity of ontology research and the generalizability of research trends to other knowledge representation methods. The further sections of the document contain our literature survey approach for identifying the state-of-the-art in section 2; we explain the broad genres identified in the ontology extraction in section 3; we proceed with a critical analysis of the major approaches through the years, by analyzing the algorithms on context relevance, efficiency and precision in section 4; we propose a deep learning based methodology (LSTM - Long Short Term Memory) to possibly overcome the gaps in the ontology enrichment in section 5 and finally end with a conclusion summarizing our observations.

2 Approach for Literature Review

We started the review on ontology learning from text before focusing on enrichment. Research on ontology learning from text started in 1995 (Mahesh et al., 1995) but still continues to be an area of interest. In the last two decades, there have been 20 survey papers on ontology tools, learning, evolution, construction, enrichment, change, generation, population, and matching with text. The large count of survey papers indicates the growing interest among researchers and changing research approaches in ontology learning. We classified these survey papers[4] on the basis of text format (structured or unstructured), evaluation methods, ontology layer cake, AI techniques, level of automation, etc. Most survey papers recommended human intervention, continued automation, gold standards and graphical interfaces for improved quality, expressiveness and scalability. While the survey papers were thorough, there weren't any papers that follow the systematic literature review (SLR) or systematic mapping process (Kitchenham, 2004), or any that discussed seminal papers that led to change of approaches.

Based on our study of the survey papers' classification methods and future directions, the keywords for search from digital libraries were ”Extraction”, ”Evolution”, ”Enrichment”, ”Maintain”, and ”Learning” along with ”Ontology” as keyword. We did not follow SLR process as our objective was to analyze the seminal papers based on context relevance, precision and algorithm efficiency. The input to our survey process consisted of 166 research papers extracted from ScienceDirect, Springer, IEEE, and ACM digital libraries from 1990-2018 time period. After reviewing the abstract and conclusion, 65 papers were eliminated from the list as they were thesis, patents, grey material, non-English, position or tutorial papers and others. The papers related to construction of data, text summarization using ontologies, machine translation, Information Retrieval, etc, of the extracted research papers were also excluded from further analysis. While there were about 23 domains for validation, Medical and Education domains were the most referred domains in the shortlisted papers. The Figure 1 (Y-axis is the count of papers and X-axis is the year of publication) on ontology learning depicts the ongoing interest of researchers. The study on approaches of the shortlisted papers stated that although natural language processing and description logic continue to be used; Word2Vec, a step towards deep learning is being more leveraged for ontology learning. The shortlisted papers were categorized after reading the abstract, introduction and conclusion, as shown in Figure 2. The papers on ”create” were related to ontology construction or population or generation. The papers on ”update” were related

[4] https://tinyurl.com/OntoSurvey

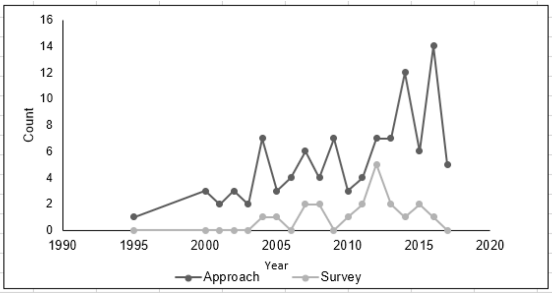


Figure 1: Trend Chart on Ontology Research

to ontology evolution, enrichment, updation, refinement, maintain, etc. The papers on ”CRUD” operation dealt with creation, updation and deletion of redundant concepts and relations as well. For further analysis, the papers on ”update” and ”CRUD” on ontology were clustered into 4 categories based on the approach used for enrichment.

1. Similarity Based Clustering Algorithms
2. Set Theoretic Based Algorithms
3. Web Corpus Based Algorithms
4. Deep Learning Based Algorithms

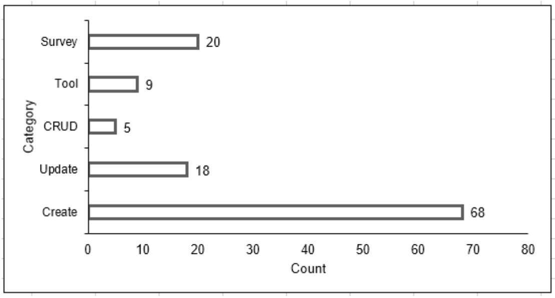


Figure 2: Ontology Learning Categories

3 Categories in Ontology Enrichment

We proceeded with a review of the 23 shortlisted papers of the 4 categories in ontology enrichment.

3.1 Similarity Based Clustering Algorithms:

Some of the earliest papers in the field of ontology enrichment from text, adopted similarity-based clustering approach. A hierarchical clustering algorithm to classify ontology-based metadata (Maedche and Zacharias, 2002) was proposed in 2002. Later, a similarity-based clustering approach was proposed to identify concepts in a gene ontology (Cheng et al., 2004). The unsupervised guided hierarchical clustering algorithm (Cimiano and Staab, 2005) uses an oracle of hypernyms derived from WordNet, text and WWW corpora for clustering concepts in a hierarchy. The fuzzy inference mechanism (Lee et al., 2007) uses fuzzy numbers that calculate the conceptual similarity between concepts to obtain new learning instances.

3.2 Set Theoretic Based Algorithms

These algorithms used a set-theoretic approach to order concepts. Harris’s distributional hypothesis (Sahlgren, 2008) modeled the context of a certain word with its dependencies, and on the basis of this information, Formal Concept Analysis (FCA) (Cimiano et al., 2005a) outputs a concept lattice which is then converted into a concept hierarchy. Also, algorithms and transformations that combine FCA and the Horn model (Ben-Khalifa and Motameny, 2007) of a concept lattice have been proposed (Haav, 2004). A fuzzy extension of FCA (De Maio et al., 2009) described an approach for automatic elicitation of ontologies by web analysis. It also formalized a method that generated an OWL-based representation of concepts, individuals and properties.

Relational Concept Analysis (RCA) (Hacene et al., 2008) constructs ontologies in a semi-automated manner by translating concept lattices with interrelated elements to concepts and relations in the ontology. RCA is an extension of FCA that allows for the processing of multi-relational datasets, each with its own set of attributes and relationships amongst themselves.

3.3 Web Corpus Based Algorithms

Web corpus Based Algorithms used web as a big data corpus to overcome problems of data sparsity. The categories and labels from Wikipedia were used to classify concepts (Cui et al., 2009; Ahmed et al., 2012; Medelyan et al., 2009) leveraging N-grams and other related NLP algorithms. The Open Linked Data (Booshehri and Luksch, 2014), a freely available source of semantic knowledge is used as a skeleton to construct ontologies (Tiddi et al., 2012). DBPedia, another crowd-sourced Linked Data dataset that extracts structured information from Wikipedia is used to enrich ontology (Booshehri and Luksch, 2015).

An automatic and unsupervised methodology that uses the Web to learn ontological concept properties, or attributes, and attribute restrictions, was proposed (Sánchez, 2010). In the ”Self Annotating Web”, globally available knowledge, or syntactic resources, were used for the creation of metadata, the basic idea being that the statistical distribution of syntactic structures on the web can be used to approximate semantics. One such algorithm that implemented this paradigm is called PANKOW (Pattern-based Annotation through Knowledge On the Web) (Cimiano et al., 2004), in which patterns were instantiated from schemata and the number of hits of related entities for each concept were counted. C-PANKOW (Cimiano et al., 2005b), or Context-driven PANKOW that outperforms its predecessor, PANKOW by downloading abstracts offline, performing linguistic analysis and using the context to resolve ambiguity.

3.4 Learning based Algorithms

In recent years, learning algorithms driven by feedback from domain experts have gained popularity. OntoAMAS (Benomrane et al., 2016) tool is based on adaptive multi-agent system (AMAS) for ontology enrichment and makes proposals based on ontologists' feedback. Also noteworth, is the Probabilistic Relational Hierarchy Extraction technique based on Probabilistic Relational Concept Extraction (Drumond and Girardi, 2010) to extract concepts and the taxonomic relationships from inference on Markov Logic Networks. Group storytelling technique has been used (Confort et al., 2015) to gather knowledge from those involved in the field in the first phase, which makes the system learn the concepts for an ontology automatically. OntoHarvester system (Mousavi et al., 2014) used deep NLP-based algorithms to mine text and extract domain-specific ontologies by iteratively extracting ontological relations that link the concepts in the ontology to the terms in the text, out of which strongly connected concepts were added to the ontology.

The Automated Ontology Generation Framework (Alobaidi et al., 2018), used Linked Biomedical Ontologies, various NLP techniques (in text processing based on ”Compute on Demand” method, N-Grams, ontology linking and classification), semantic enrichment (using RDF mining), syntactic pattern and graph-based techniques (to extract relations), and domain inference engine (to build the formal ontology). They also proposed Linked Biomedical Ontologies as a promising solution towards automating the ontology generation process in the disease-drug domain. Word2Vec was used (Wohlgenannt and Minic, 2016) to extract similar meaning terms or concepts and to get certain semantic and syntactic relations based on simple vector operations. The word representations derived from traditional Distributional Semantic Models such as Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) assume that words in similar contexts have similar embeddings. Word embeddings using neural language models, for example, CBOW and Skip gram, begin usage of deep learning. (Casteleiro et al., 2016) focused on the performance of LDA, LSA, Skip gram and CBOW algorithms in ontology enrichment.

4 From Clustering to Learning algorithms: An in-depth analysis

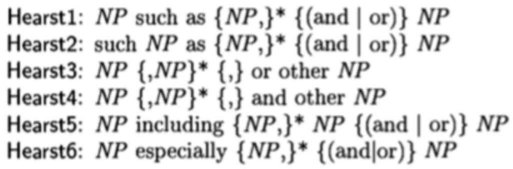
After categorizing our research set of 23 papers into 4 categories, the seminal algorithms from each category are listed below:

1. Similarity Based Clustering Algorithms: Guided Agglomerative Clustering (Cimiano and Staab, 2005)
2. Web Corpus Based Algorithms: C-PANKOW Algorithm (Cimiano et al., 2005b)
3. Set Theoretic Based Algorithms: Constructing a concept hierarchy using Formal Concept Analysis (Cimiano et al., 2005a)
4. Deep Learning Based Algorithms: The Word2Vec-based algorithm (Wohlgenannt and Minic, 2016)

In this section, we performed an in-depth analysis of these algorithms and compared their performance based on ontology evaluation (Netzer et al., 2009) methods like contextual relevance, precision and algorithmic efficiency.

4.1 Guided Agglomerative Clustering

The guided agglomerative clustering algorithm (Cimiano and Staab, 2005) has a citation count of 99 and published in 2005. The paper is based on Harris's distributional hypothesis and works by clustering concepts based on their similarities. Hypernym oracle extracted using different methods is the driving factor in the clustering process. Hypernyms oracle is constructed with the help of Hearst Patterns (Hearst, 1992). Hearst Patterns 3, used Noun Phrases (NPs) consisting of a determiner, an optional adjective sequence and a common noun sequence which constitutes the NP head. The hypernym oracle H(t) is constructed using the following three sources:



Hearst1: *NP* such as {*NP*,}* {(and | or)} *NP*
Hearst2: such *NP* as {*NP*,}* {(and | or)} *NP*
Hearst3: *NP* {,*NP*}* {,} or other *NP*
Hearst4: *NP* {,*NP*}* {,} and other *NP*
Hearst5: *NP* including {*NP*,}* *NP* {(and | or)} *NP*
Hearst6: *NP* especially {*NP*,}* {(and|or)} *NP*

Figure 3: Hearst Patterns (Cimiano and Staab, 2005)

1. WordNet: Uses synsets from WordNet for extracting hypernyms
2. Text Corpus using Hearst Patterns: Hearst Patterns were matched against the underlying text corpus, by using a regular expression comprising of POS tags to match Noun Phrases, thus constructing an is-a relation between the two terms.
3. WWW Corpus using Hearst Patterns: Every concept of interest is instantiated in a Hearst Pattern to form queries to Google API, and the abstracts from the results were downloaded offline. Hearst patterns were matched against these abstracts similar to how they were matched in the text corpus, and is-a relations were extracted accordingly.

The algorithm takes a list of words to be clustered as input. Once the hypernym oracle was constructed, each of these terms were paired up and sorted in the descending order of similarity. The clustering algorithm used the oracle to construct parent-child or sibling relationships between these terms. After this step, the unclassified terms were classified using the r-matches relation.

Though WordNet provides easy and accurate hypernyms, it is not extensive and has a very limited scope. It does not classify proper nouns or infrequently occurring terms, leading to most instances remaining unclassified leading to sparsity and scalability issues. Moreover, matching Hearst Patterns had very bad precision (13%), as shown in Figure 4 and outputs a lot of noisy data. This is due to the algorithm paying no attention to context relevance and extracting hypernyms that were irrelevant to a domain. The same word that had different meanings in different contexts (for instance, bank - which could refer to a river bank or a blood bank or a financial bank) were clustered together. In addition, this approach disregarded a lot of relevant relations because it relied on an exact syntactic pattern match that pays no attention to semantics.

4.2 C-PANKOW algorithm

The C-PANKOW algorithm (Cimiano et al., 2005b) again by Cimiano et al. has a citation count of 246. The algorithm was based on the paradigm of ”Learning by Googling”. In this paradigm, given an instance, evidence was collected from the internet for the possible concepts. Then, either the instance was mapped to the concept with maximum evidence, or alternatively, an engineer with domain-specific knowledge does mapping manually. The PANKOW (Pattern-based Annotation through Knowledge on the Web) algorithm (Cimiano et al., 2004), the predecessor of the C-PANKOW algorithm, instantiated a query using pre-defined patterns or regular expressions. A one-to-one mapping was done between each concept and instance to generate a query from these patterns. This query, similar to how the hypernym oracle was extracted using the WWW corpus in Guided Clustering, was made available to the Google API and the number of hits for this query were counted. Based on the statistical web fingerprint, or the total number of search results for each entity, the instance were mapped to the concept to get *disambiguation by maximal evidence*. The statistical web fingerprint were presented to the knowledge engineer to review and take the final decision. However, PANKOW had a few disadvantages. Firstly, it issued a large number of requests to the Google Web API, which is proportional to the number of ontology concepts, so it does not scale well for large ontologies. Also, because of the restrictions inherent in the generation of patterns, many actual instances were not found.

C-PANKOW addresses some issues by downloading results of queries, or the abstracts, and then doing the pattern matching locally by linguistic analysis. Downloading web pages reduced the number of requests made to Google Web API and the network traffic by issuing a constant number of queries per instance. In addition, it factors context into consideration and calculates the contextual similarity between two pages before doing concept-instance mapping, which reduced ambiguity especially in cases where a word has multiple meanings and its meaning depends on context. C-PANKOW presented a novel idea to concept extraction by combining the approaches of maximum frequency-based mapping and document similarity-based filtering. Frequency-based mapping reduces noise and gives only the most relevant relations, whereas similarity-based filtering using Doc2Vec (Lau and Baldwin, 2016) helps partially address the issue of context relevance by preemptively filtering out irrelevant abstracts. These two approaches augmented C-PANKOW's precision (36%) to be more than that of Guided Clustering. The filtering also increased algorithmic efficiency as, unlike Guided Clustering, it does not look for matches in irrelevant documents. However, despite its advantages, since C-PANKOW (like Guided Clustering) uses naive syntactic pattern matching to extract hypernymy relations, it does yield noisy data as well, whilst ignoring relevant results. This is because: a) The pattern matching fails to take semantics and language structure into consideration. b) It is also ineffective in situations where the concept being referred were already defined in an earlier sentence c) Though Doc2Vec does partially address the issue of context relevance at the document level, it does not check the relevance at the sentence or paragraph level, resulting in noisy data as well.

To address the concerns of disambiguation in concepts or relations, agent based models have been proposed for the enrichment of ontologies (Sellami et al., 2013). An agent has local knowledge about itself and other neighbour agents, as well as about the lexical terms and concepts extracted from the corpus. It uses this knowledge to evaluate its own relevance in the ontology and manage its relationships with other agents. When new documents are added to the corpus, or when the ontologist suggests changes to the ontology proposed by the MAS, there were perturbations or disturbances caused in the system. Each agent in the MAS reacts to these perturbations by modifying its relations with other agents, updating its knowledge on and/or communicating with other agents in order to reach a stable state. On reaching this stable state, the MAS proposes a new version of the ontology which is once again presented to the ontologist. The ontologist suggests changes again and this whole process continues iteratively till the MAS reaches a final state where the ontology is not challenged by him anymore. In DYNAMO-MAS (Sellami et al., 2013) word disambiguation is handled by the Teminological Ontological Resource (TOR) model which comprises of a conceptual component (the ontology) and a lexical component (the terminology). Terms were attached to concepts by denotation links and contain a confidence score. These denotation links can be changed by the agents if a request with a higher confidence score is made. Thus, any term is attached by a denotation link to the concept with the highest confidence score. Since the same term can have different meanings in different context, the TOR model is able to disambiguate the meaning using these confidence scores. However, the confidence score is partly generated from a pattern score, which in turn has to be manually defined from empirical evaluations. Moreover, the ontologist has to manually verify the annotations proposed by the MAS which in turn means the text corpus has to be limited to a few hundred documents and cannot work on the larger web corpus. Thus while this approach makes a massive progress towards solving the issue of context relevance, it suffers from scalability issues.

4.3 Constructing concepts using FCA

(Cimiano et al., 2005a) has 693 citations and is the primary source for research on FCA from text corpora. The algorithm was based on Set-Theoretic approach that uses FCA to convert a partial order to a concept hierarchy on the basis of syntactic dependencies taken as features. With NLTK, the Part of Speech (POS) tags are extracted, separated into chunks, reduced to base-form (lemmatized), smoothed to overcome data sparseness, weighted, and only those terms with values above a threshold are converted into a formal context (Ganter and Wille, 1999). FCA (Ganter and Wille, 1996) is then applied to this context to transform into a partial order, which is then compacted to remove abstract concepts and get the final concept hierarchy. This algorithmic approach uses pseudo-syntactic dependencies to extract concepts from the parse tree. Hence, it significantly outperforms Guided Clustering and C-PANKOW in terms of precision. This algorithm forms clusters and also provides an intentional description for them, leading to better understanding. However, this algorithm does not identify labels that describe the intention of a specific cluster, resulting in sparsely populated concepts. In addition, it is inefficient as construction of a separate concept lattice for every document is time expensive. Thus while it is more efficient than Clustering, it loses out to C-PANKOW in efficiency. However, the greater precision does shows that enriching contextual features using pseudo-syntactic dependencies is a viable alternative that outperforms enriching from parse trees.

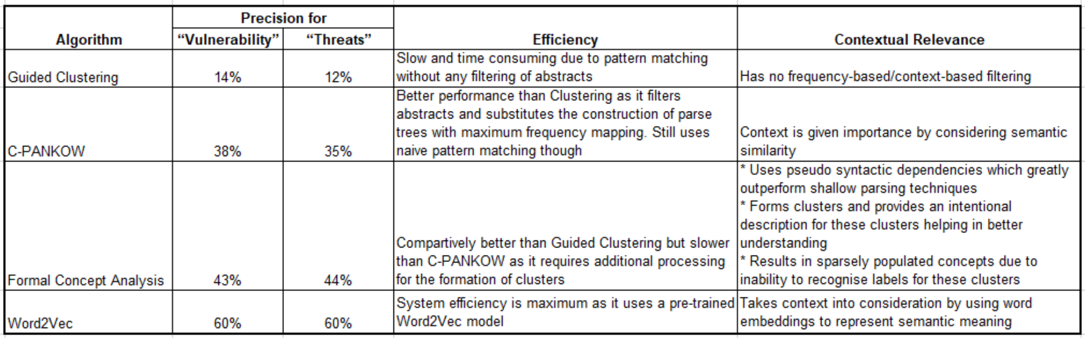
4.4 Word2Vec-based algorithm

Word2Vec, a 2-layer neural network has also been used to build a sample ontology learning system (Wohlgenannt and Minic, 2016). The neural nets are trained on the linguistic context by Word2Vec, using two methods: Continuous Bag of Words (CBoW) and skip grams. CBoW is used to predict the context of a word, given the word, while skip grams predict the context given the word as input. Word2Vec allows vector operations, and is trained to output high quality similar terms given any input term. The Word2Vec model can be trained on the Google News corpus on any other large corpus.

The algorithm provided higher percentage of relevant concepts that can be used to enrich the ontology. In addition to having greater precision (60%) and efficiency than the previous algorithms, this algorithm makes headway in solving the issue of contextual relevance by using CBoW and Skip Grams to train the model. However, it does have a few drawbacks. Firstly, for terms that aren't encountered by the model in training corpus, a word embedding is not constructed, hence, concepts remaining unclustered. Secondly, Word2Vec doesn't have any shared representations at sub-word levels. It represents each word as an independent vector, though there could be morphologically similar terms. It also detects concepts that are too close to the original term, like plurals and synonyms which are unneccessarily added to the ontology as separate concepts. Lastly, it necessitates manual intervention after every iteration, unlike the previous algorithms, which in turn means it suffers from scalability issues.

5 Discussion

The Guided Agglomerative Clustering algorithms used Wordnet and Hearst Patterns on corpora to build its hypernym oracle. While Wordnet is able to provide hypernyms for common nouns, it cannot handle proper nouns and phrases, which are often the primary focus while enriching domain specific ontologies. Using Hearst Patterns is inefficient too and results in a lot of noise, due to pattern being matching being purely syntactic with no attention paid to context. Though C-PANKOW is able to improve on precision, efficiency and also partially address the issue of context relevance (using a mixture of frequency-based mapping and document similarity scores), it uses naive syntactic pattern matching which results in selecting irrelevant terms and dropping relevant ones. The DYNAMO-MAS algorithm, despite solving disambiguation and having better precision, has serious limitations like data sparsity and unscalability. FCA, which uses pseudo-syntactic dependencies, was found to have better precision than both Clustering and C-PANKOW. But construction of a concept hierarchy is time inefficient, which is where it loses out to C-PANKOW. The Word2Vec algorithm was able to improve the problems of efficiency, precision and data sparsity by using word embeddings and skip-grams, and was found to outperform previously mentioned algorithms. However, this algorithm also suffers from some shortcomings like the inability to handle previously unencountered words, selecting of too similar terms, scalability issues due to manual intervention etc. We used the 'Information Security' ontology (Ekelhart et al., 2006) based on ISO 27001 for comparing the algorithms. Figure 4 shows comparison of the metrics across these algorithms. All these algorithms have an area of improvement when the current concept and its pronouns are being extracted from text. In the previous approaches, the attributes and relations were mapped to the *pronouns* and not



Algorithm	Precision for		Efficiency	Contextual Relevance
	“Vulnerability”	“Threats”		
Guided Clustering	14%	12%	Slow and time consuming due to pattern matching without any filtering of abstracts	Has no frequency-based/context-based filtering
C-PANKOW	38%	35%	Better performance than Clustering as it filters abstracts and substitutes the construction of parse trees with maximum frequency mapping. Still uses naive pattern matching though	Context is given importance by considering semantic similarity
Formal Concept Analysis	43%	44%	Compartively better than Guided Clustering but slower than C-PANKOW as it requires additional processing for the formation of clusters	* Uses pseudo syntactic dependencies which greatly outperform shallow parsing techniques * Forms clusters and provides an intentional description for these clusters helping in better understanding * Results in sparsely populated concepts due to inability to recognise labels for these clusters
Word2Vec	60%	60%	System efficiency is maximum as it uses a pre-trained Word2Vec model	Takes context into consideration by using word embeddings to represent semantic meaning

Figure 4: Ontology Algorithms Comparison

the concept itself. To address this gap, the algorithm needs to retain in memory the concept being extracted, instead of using naive pattern-matching approaches. Also, the analysis of the shortlisted research papers in each category state the declining research on Clustering and Set-Theoretic algorithms, and an increasing in research of learning algorithms.

5.1 Possible solution: Long Short Term Memory Networks

The algorithms proposed above use either pattern matching techniques, naive SVO (subject-verb-object) triplet extraction techniques or semantic similarity techniques for extracting concepts. All of these techniques were at the concept level, and though extension of algorithms like C-PANKOW used Doc2Vec to gauge similarity of documents, none of these algorithms involved understanding of the text corpus to filter out irrelevant data. Hence, we suggest a need to incorporate Deep Learning to enrich ontologies.

We propose a Deep Learning solution using Long Short Term Memory Networks (LSTMs) [5] to address the identified gaps. We explain our reason for proposing an LSTM with the help of an example.

”Cross-Frame Scripting (XFS) is a browser based attack that combines malicious JavaScript with an iframe while loading a legitimate site. This attack is one of the most common attacks against IE. This is due to it leaking keyboard events across HTML framesets.”.

On passing these sentences to a concept-relationship extraction system, such as the ones described previously, a ”one-of” relationship would be formed between ”This attack” and ”one of the most common attacks against IE” and a ”due-to” relationship would be formed between ”this” and ”leaking keyboard events across HTML framesets”. However, in the second sentence, ”This attack” refers to ”XFS attack” (from the first sentence) and is the concept identified and can be abstracted to ”browser based attack”. But in the third sentence, the current concept has changed and ”this” refers to ”attacks against IE”. Hence, normal concept extraction techniques would not work for these examples, since the current concept may change every sentence. LSTMs can be trained to learn optimal forget matrices that continually update the cell state, thereby, enabling the model to maintain the state of a concept (by addding new concepts and removing old ones) for longer durations. Thus, LSTMs can enable greater semantic understanding as well as detection of long ranging patterns, which theoretically should improve precision.

6 Conclusion

We started this survey paper by describing the need for enrichment of ontologies. We proceeded to survey the existing domain literature in the field of ontology learning from text and got a subset of 166 research papers and 20 survey papers. From shortlisted 101 papers, we narrowed down to the 23 most relevant research papers. These 23 papers were classified into four categories based on the approach used for ontology enrichment, namely Clustering, Set-Theoretic, Web Corpus-based and Learning-based Algorithms. We selected a seminal paper from each category, based on criteria like the date of publication, the number of citations, relevance to our end goal etc. and then described the approach of the algorithms. Next,

[5] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

we compared algorithms performance (context relevance, precision and efficiency) on the enrichment of Information Security ontology. We found that with each trend, some of the gaps were overcome but there still remained the problem of retaining concepts to improve relevance in a scalable manner. We proposed LSTMs as a possible solution for concept retention, since they use a memory state to partially remember/forget concepts over long periods of time as require. In future, we plan on implementing the proposed LSTM model to improve precision and efficiency of the state-of-the-art. We also plan to validate further with complex ontologies, and extend our concept enrichment model to the addition of instances for building knowledge base.

References

Khalida Bensidi Ahmed, Adil Toumouh, and Mimoun Malki. 2012. Effective Ontology Learning: Concepts' Hierarchy Building using Plain Text Wikipedia. In *ICWIT*, pages 170–178.

Mazen Alobaidi, Khalid Mahmood Malik, and Maqbool Hussain. 2018. Automated Ontology Generation Framework Powered by Linked Biomedical Ontologies for Disease-Drug Domain. *Computer Methods and Programs in Biomedicine*.

Kamel Ben-Khalifa and Susanne Motameny. 2007. Horn-Representation of a Concept Lattice. volume 38.

Souad Benomrane, Zied Sellami, and Mounir Ben Ayed. 2016. An ontologist Feedback driven Ontology Evolution with an Adaptive Multi-agent System. *Advanced Engineering Informatics*, 30(3):337–353.

Meisam Booshehri and Peter Luksch. 2014. Towards Adding Linked Data to Ontology Learning Layers. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services*, pages 401–409. ACM.

Meisam Booshehri and Peter Luksch. 2015. An Ontology Enrichment Approach by using DbPedia. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, page 5. ACM.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. 2005. Ontology Learning from Text: An overview. *Ontology Learning from Text: Methods, Evaluation and Applications*, 123:3–12.

Mercedes Argüello Casteleiro, Maria Jesus Fernandez Prieto, George Demetriou, Nava Maroto, Warren J Read, Diego Maseda-Fernandez, Jose Julio Des Diz, Goran Nenadic, John A Keane, and Robert Stevens. 2016. Ontology Learning with Deep Learning: a Case Study on Patient Safety Using PubMed. In *SWAT4LS*.

Jill Cheng, Melissa Cline, John Martin, David Finkelstein, Tarif Awad, David Kulp, and Michael A Siani-Rose. 2004. A Knowledge-based Clustering Algorithm Driven by Gene Ontology. *Journal of Biopharmaceutical statistics*, 14(3):687–700.

Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. 2004. Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web*, pages 462–471. ACM.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005a. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339.

Philipp Cimiano, Günter Ladwig, and Steffen Staab. 2005b. Gimme'the Context: Context-driven Automatic Semantic Annotation with C-PANKOW. In *Proceedings of the 14th international conference on World Wide Web*, pages 332–341. ACM.

Philipp Cimiano and Steffen Staab. 2005. Learning Concept Hierarchies from Text with a Guided Agglomerative clustering Algorithm. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*.

Valdemar TF Confort, Kate Revoredo, Fernanda Araujo Baião, and Flávia Maria Santoro. 2015. Learning Ontology from Text: A Storytelling Exploratory Case Study. In *International Conference on Knowledge Management in Organizations*, pages 477–491. Springer.

Gaoying Cui, Qin Lu, Wenjie Li, and Yirong Chen. 2009. Mining Concepts from Wikipedia for Ontology Construction. In *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 03, pages 287–290. IEEE.

Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Sabrina Senatore. 2009. Towards an Automatic Fuzzy Ontology Generation. In *IEEE International Conference on Fuzzy Systems*, pages 1044–1049. IEEE.

Lucas Drumond and Rosario Girardi. 2010. An Experiment using Markov Logic Networks to Extract Ontology Concepts From Text. *lLearning*, 1:2.

Andreas Ekelhart, Stefan Fenz, Markus D Klemen, and Edgar R Weippl. 2006. Security Ontology: Simulating Threats to Corporate assets. In *International Conference on Information Systems Security*, pages 249–259. Springer.

Andreas Faatz and Ralf Steinmetz. 2002. Ontology Enrichment with Texts from the WWW. *Semantic Web Mining*, 20.

Bernhard Ganter and Rudolf Wille. 1996. Formal Concept Analysis. *Wissenschaftliche Zeitschrift-Technischen Universitat Dresden*, 45:8–13.

Bernhard Ganter and Rudolf Wille. 1999. Contextual Attribute Logic. In *International Conference on Conceptual Structures*, pages 377–388. Springer.

Hele-Mai Haav. 2004. A Semi-automatic Method to Ontology Design by Using FCA. In *CLA*. Citeseer.

Mohamed Rouane Hacene, Amedeo Napoli, Petko Valtchev, Yannick Toussaint, and Rokia Bendaoud. 2008. Ontology Learning from Text using Relational Concept Analysis. In *International MCETECH Conference on e-Technologies*, pages 154–163. IEEE.

Marti A Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th conference on Computational linguistics*, volume 2, pages 539–545. Association for Computational Linguistics.

Barbara Kitchenham. 2004. Procedures for Performing Systematic Reviews. *Keele, UK, Keele University*, 33(2004):1–26.

Jey Han Lau and Timothy Baldwin. 2016. An Empirical Evaluation of Doc2Vec with Practical Insights into Document Embedding Generation. *arXiv preprint arXiv:1607.05368*.

Chang-Shing Lee, Yuan-Fang Kao, Yau-Hwang Kuo, and Mei-Hui Wang. 2007. Automated Ontology Construction for Unstructured Text Documents. *Data & Knowledge Engineering*, 60(3):547–566.

Alexander Maedche and Valentin Zacharias. 2002. Clustering Ontology-based Metadata in the Semantic Web. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 348–360. Springer.

Kavi Mahesh, Sergei Nirenburg, et al. 1995. A Situated Ontology for Practical NLP. In *Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, volume 19, page 21. Citeseer.

Olena Medelyan, David Milne, Catherine Legg, and Ian H Witten. 2009. Mining Meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754.

Hamid Mousavi, Deirdre Kerr, Markus Iseli, and Carlo Zaniolo. 2014. Harvesting Domain Specific Ontologies from Text. In *IEEE International Conference on Semantic Computing*, pages 211–218. IEEE.

Yael Netzer, David Gabay, Meni Adler, Yoav Goldberg, and Michael Elhadad. 2009. Ontology Evaluation through Text Classification. In *Advances in Web and Network Technologies, and Information Management*, pages 210–221. Springer.

Natalya F Noy and Michel Klein. 2004. Ontology Evolution: Not the same as Schema Evolution. *Knowledge and Information Systems*, 6(4):428–440.

Magnus Sahlgren. 2008. The Distributional Hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

David Sánchez. 2010. A Methodology to Learn Ontological Attributes from the Web. *Data & Knowledge Engineering*, 69(6):573–597.

Zied Sellami, Valérie Camps, and Nathalie Aussenac-Gilles. 2013. DYNAMO-MAS: a Multi-agent System for Ontology Evolution from Text. *Journal on Data Semantics*, 2(2-3):145–161.

Ilaria Tiddi, Nesrine Ben Mustapha, Yves Vanrompay, and Marie-Aude Aufaure. 2012. Ontology Learning from Open Linked Data and Web Snippets. In *Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 434–443. Springer.

Gerhard Wohlgenannt and Filip Minic. 2016. Using word2vec to Build a Simple Ontology Learning System. In *International Semantic Web Conference (Posters & Demos)*.