Creating and Evaluating Data-Driven Ontologies

Maaike H.T. de Boer and Jack P.C. Verhoosel

Data Science department, TNO (Netherlands Organisation for Applied Scientific Research),
Anna van Buerenplein 1, 2595 DA, The Hague, The Netherlands
Email: maaike.deboer@tno.nl and jack.verhoosel@tno.nl

***Abstract*—Automatically creating data-driven ontologies is a challenge but it can save time and resources. In this paper, eight data-driven algorithms are compared to create ontologies, four ontologies based on documents and four based on keywords, on three different document sets. We evaluate the performance using three different evaluation metrics based on nodes, weights and relations. Results show that 1) keyword-based methods are in general better than document-based methods; 2) a co-occurrence-based algorithm is the best document-based method; 3) the evaluation metrics give useful insight, but need to be enhanced in future work. It is advised to a) use the created ontologies as a head start in an ontology creation session, but not use the ontologies as created; b) use word2vec to generate an ontology in a generic domain, whereas the co-occurrences algorithm should be used in specific domain.**

Keywords–Ontologies; Machine Learning; NLP; Word2vec; Ontology Learning; F1 score.

I. Introduction

In the previous decade, data scientists often used either a knowledge-driven or a data-driven approach to create their models / classifiers. In the knowledge-driven approach, the (expert) knowledge is structured in a model, such as an ontology. Some of the advantages of this type of approach are that it is insightful for humans, validated by experts, and it gives a feeling of control. Some of the disadvantages of the knowledge-driven approach are that it takes a lot of dedicated effort to construct the model, it is hard to provide the full model (this is only possible in closed-world domains) and that there might not be one truth. For example, if two experts separately create a knowledge model about the same domain, they probably will come up with different ones, because each expert has his/her own subjective view of important concepts and relations in the domain. Data-driven approaches do not need the dedicated effort from people to construct the model, because an algorithm is used that extracts a model much faster. Disadvantages of data-driven approaches are that the models are often not insightful for humans, they might contain too much noise and might be less 'crisp'. As knowledge-driven and data-driven approaches each have their advantages, a combination of both approaches is worthwhile to use. The field in which ontologies learn from available knowledge using data is named *ontology learning*.

This paper is an extension of our previous paper on this topic [1]. In our previous paper, we proposed an ontology learning methodology that uses existing and new data-driven algorithms to create ontologies based on unstructured textual documents in the agriculture domain. In this paper, we broaden our scope to the pizza domain. Whereas evaluation in the agriculture domain is harder because our document set did not have a matching ground truth ontology, we use the well-known pizza ontology [2] to validate our created initial ontology in that domain. We also use the same pizza document set as presented by Rospoucher et al. [3] to make a comparison possible. Additionally, we extracted another pizza document set based on Wikipedia. In our experiments, eight different ontologies are created for each document set and the performance is evaluated using three evaluation metrics, of which two are those proposed by Rospoucher et al. [3] and our previously proposed relation-based evaluation metric.

In the next section, the related work is described on ontology learning, including open information extraction, and the evaluation of ontologies. In Section III, the methods used to create the different ontologies are explained. Section IV describes the experimental set-up with the datasets, characteristics of the resulting ontologies and our evaluation methodology. Section V contains the results of the evaluation and Section VI contains the discussion of our results. Finally, Section VII contains the main conclusions as well as a description of future work.

II. Related Work

Ontology learning is focused on learning ontologies based on data [4] [5]. One of the most known concepts in ontology learning is the ontology learning layer cake. Starting from the bottom of the cake, the order from bottom to the top of the layer is terms, synonyms, concept formation, concept hierarchy, relations, relation hierarchy, axiom schemata and finally general axioms. Similar to the layered cake, Gillani et al. [6] describe the process of ontology learning by input, term extraction, concept extraction, relation extraction, concept categorization, evaluation, ontology mapping. Ontologies can be learned in three kind of strategies: structured, semi-structured and unstructured data [4]. Examples of these different strategies are: database (structured), HTML or XML (semi-structured) and texts (unstructured). Besides the learning strategies, there are three types of tools available: ontology editing tools, ontology merging tools and ontology extraction tools [7]. In this paper, the focus in on automatically creating ontologies from text, so we focus on unstructured data and ontology extraction tools.

A. OpenIE

Some available ontology extraction tools only focus on the information extraction, up to the relation extraction part of the layered cake. This means that these tools only focus on the creation of triples with a subject, verb or relation, and object. The field that focuses only on the creation of these triples is named Open Information Extraction (OpenIE). According to a recent systematic mapping study by Glauber and Claro [8], the two main steps in OpenIE methods are: 1) shallow analysis or dependency analysis for sentence annotation, such as Part of Speech (PoS) tagging or using the Stanford Dependency Parser; 2) machine learning or handcrafted rules for the extraction of relationship triples. Niklaus et al. [9] make the division between learning-based systems, rule-based systems, clause-based systems and system capturing inter-propositional relationships.

One of the first OpenIE tools is TextRunner [10]. TextRunner tags sentences with PoS tags and noun phrase chunks, in a fast manner with one loop over all documents. TextRunner was followed by WOE (pos and parse), ReVerb, KrakeN, EXEMPLAR, OLLIE, PredPatt, ClausIE, OpenIE4, CSD-IE, NESTIE, MinIE and Graphene among others [8], [9]. All methods use a combination of the two main steps mentioned above. For example, WOE [11] uses machine learning on Wikipedia to learn extraction patterns with PoS tags and dependency parsers. REVERB [12] uses syntactic constraints in the form or PoS-based regular expressions to reduce the number of incoherent and uninformative extractions. OLLIE [13], a follow-up from REVERB, learns from a training set the extraction pattern templates using dependency parsers. It also uses contextual information by adding attribution and clausal modifiers. Most methods often solve the problem of increasing informativeness or decreasing computational complexity [8]. Informativeness links to the number of relevant facts. This is often tackled in the second step, either by increasing the facts using co-reference or transitive inference such as in ClausIE [14], or reducing the facts by using lexical constraints such as in REVERB [12]. Many of the OpenIE tools are not fast and very computational expensive. WOEpos [11] is for example 30 times faster than the original WOEparser, but less accurate.

Recently, deep learning methods, such as the encoder-decoder framework from Cui et al. [15], and the relation extraction method from Lin et al. [16] have been proposed. Although these methods seem fruitful, deep learning seems not yet as overwhelmingly better in all tasks within open information extraction as compared to the field of computer vision [17].

Related to the OpenIE field, query expansion can also be used to find more concepts and relations [18]. This method is often used in the information retrieval field. The most common method is to use WordNet [19]. Boer et al. [20] [21] also use ConceptNet to find related concepts and their relations. Word2vec is also used in information retrieval [22] and ontology enrichment [23] [24].

Concluding, the OpenIE field is quite advanced with many tools and techniques. In our paper, we use some of the state of the art techniques, and use the state of the art from the query expansion field and apply it in the OpenIE field.

B. Ontology Learning tools

One of the oldest methods that use the full ontology learning layered cake is Terminae [25]. Terminae is a method and platform for ontology engineering, and includes linguistic analysis with Natural Language Processing (NLP) tools to extract and select terms and relations, conceptual modeling / normalization (differentiation, alignment and restructuring) and formalization / model checking, with the syntactic and semantic validation.

A second tool is OntoLT [26], which is available as a plugin in Protégé and enables mapping rules. Linguistic annotation of text documents is done using Shallow and CHunk-based Unication Grammar tools (SCHUG) [27], which provide annotation of PoS, morphological inflection and decomposition, phrase and dependency structure. The mapping rules can then be used to map the ontologies or the document into one ontology.

A third tool is Text2Onto [28]. Text2Onto uses GATE to extract entities. GATE [29] has a submodule named ANNIE that contains a tokeniser, sentence splitter, PoS tagger, gazetteer, nite state transducer, orthomatcher and coreference resolver. Several metrics, such as Relative Term Frequency (RTF), Term Frequency Inverted Document Frequency (TF-IDF), Entropy and the C-value/NC-value are used to assess the relevance of a concept. The relations between concepts are found with WordNet, Hearst patterns, and created patterns in JAPE. With the Probabilistic Ontology Model, the tool should be robust to different languages and changing information. According to Zouaq et al. [30], Text2Onto generates very shallow and light weight ontologies.

A fourth tool is Concept-Relation-Concept Tuple based Ontology Learning (CRCTOL) [31]. CRCTOL uses the Stanford PoS tagger and the Berkeley parser to assign syntactic tags to the words. They use a Domain Relevance Measure (DRM), a combination of TF-IDF and likelihood ratio, to determine the relevance of a word or multi-word expression. LESK and VLESK are used for word sense disambiguation. Hearst patterns, relations in WordNet and created patterns with regular expressions are used to find relations with the relevant terms. According to Gillani et al. [6], CRCTOL only creates general concepts and ignores whole-part relations, the ontology is not the comprehensive and accurate representation of a given domain and it is time-consuming to run the tool, because it does full-text parsing.

A fifth tool is CFinder [32], which is created to automatically find key concepts in text. They use the Stanford PoS tagger, a dictionary lookup for synonym finding, stopword removal, and combination of words to also have dependent phrases as concepts. The key concepts are then extracted using a rank-based algorithm that uses the TF (Term Frequency) and a domain specific DF (Document Frequency) as weight. The paper stops at the key concept extraction and does not go further with determining relations.

A sixth tool is OntoUPS [33]. OntoUPS uses the Stanford dependency parser, and learns an Is-A hierarchy over clusters of logical expressions, and populates it by translating sentences to logical form. It uses Markov Logical Networks (MLNs) for that.

A seventh tool is OntoCMaps [30], which uses the Stanford PoS tagger and dependency parser to extract concepts. It uses several generic patterns to extract relations.

A eighth tool is Promine [6]. Promine uses tokenization, stop word filtering, lemmatization, and term frequency to create a set of key words. Wordnet, Wiktionary and a domain glossary (AGROVOC) are used for concept enrichment. The relevance, or term goodness, is calculated with the information gain, which combines the entropy and conditional probability. The concepts are filtered using the information gain, path length and depth of concepts.

Besides, FRED [34] transforms text to LinkedData, using theory from combinatory Categorial Grammar, Discourse Representation Theory, Frame Semantics and Ontology Design Patterns.

Additionally, Tiddi et al. [35] use LinkedData as a basis to create an ontology. They use a dependency analysis to extract entities and the TF-IDF frequency to filter patterns. An entity discovery is done using web queries.

Bendaoud et al. [36] have a semi-automated process in which an ontology is constructed from document abstracts. The formal concept analysis framework is extended to a relational

concept analysis to find links and infer relations between concepts.

Related to ontology learning, Mittal et al. [37] recently combined knowledge graphs and vector spaces into a VKG structure. In that way, both a smart inference from the knowledge graphs and a fast look-up from the vector spaces are combined. This method, however, does not automatically create a new ontology from text documents.

Also, deep learning is used in knowledge graphs. Schlichtkrull et al. [38] propose a Graph Convolutional Network to predict missing facts and missing entity attributes. This method can, thus, also not create an ontology from a set of documents, but is able to enrich an existing ontology.

Concluding, many ontology learning tools are already available and many use OpenIE techniques first and build upon those. We use some of the OpenIE used in the tools as state of the art for our algorithms.

C. Evaluating ontologies

Brank et al. [39] state that most approaches to evaluate ontologies can be placed in one of the following categories:

- Golden Standard: compare to "golden standard"
- Application-based: use in application and evaluate results
- Data-driven: involve comparisons with a data source
- Assesment by humans: human evaluation based on a set of predefined criteria, standards, and / or requirements

Hlomani et al. [40] also use these approaches in their survey, and state the advantages and disadvantages of each approach. We focus on the disadvantages of the approaches first. In the golden standard, the main disadvantage is the evaluation of the golden standard and the performance is highly dependent on the quality of the golden standard. In the application-based approach, the disadvantage is generalizability: what might be a good ontology in one application does not have to be a good one in another. The application-based approach is also only applicable for a small set of ontologies. The main disadvantage of the data-driven approach is that the domain knowledge is assumed to be constant, which is not the case. Finally, the disadvantage of the human assessment is subjectivity.

In this paper, we focus on the data-driven evaluation a well as a golden standard for one of our domains. In the data-driven approach the ontology is often compared against existing data about the domain. Many papers on this topic focus on some kind of coverage of the domain knowledge within the ontology [41]–[44]. For example, Brewster et al. [44] compare extracted terms and relations from text with the concepts and relations in an ontology. They use a probabilistic model to determine the best ontology for a certain domain. OOPS! focusses on pitfalls in ontologies and target newcomers and domain experts [45].

Besides the categories, ontologies can be evaluated on different levels. These levels are defined differently in different papers. Brank et al. [39] divide the levels in lexical, hierarchical, other semantic relations, context, syntactic, and structure. They link the categories and the levels in a matrix, in which the human assessment is the only category which evaluates on all levels. The data-driven approach can only evaluate on the first three levels. The distinction of Burton et al. [46] is syntactic, semantic, pragmatic and social. Gangemi et al. [47] use the distinction between structural, functional and usability-profiling. Burton et al. [46] use lawfulness, richness, interpretability, consistency, clarity, comprehensiveness, accuracy, relevance, authority, and history. Lozano et al. [48] even use a three-level framework of 117 criteria. Hlomani et al. [40] make the distinction between ontology quality and ontology correctness views on ontology evaluation. For ontology quality, they focus on computational efficiency, adaptability and clarity. Ontology correctness uses accuracy, completeness, conciseness and consistency. Tiddi et al. [35] use the F-measure and precision and recall to evaluate ontology correctness by checking 1) whether attribute values are correctly extracted and 2) how much of the existing knowledge is extracted (based on DBpedia). Rospocher et al. [3] use the same performance metrics to compare an ontology with a list of automatically extracted keywords. Recently, Mcdaniel et al. [49] introduced the DOORS framework in which ontologies can be ranked by using syntactic, semantic, pragmatic and social quality metrics.

III. Method

In this paper, we create a taxonomy or concept hierarchy, and we do not include the top two layers of the layered cake (domain, range and axioms / generic rules). This means that this is a first step towards an ontology, but although we have created an owl file, it is not as rich as a real ontology with domains and rules. Figure 1 shows an overview of the methods used to create the ontologies. From each article first the plain text is extracted from the PDF. On these plain texts sentence splitting is used, as well as tokenizing, removing non-ascii and non-textual items and non-English sentences as pre-processing. With these pre-processed texts the ontologies named Hearst, Co-oc and OpenIE (explained below), an our previously proposed Dep++ method [1] are created. The ontologies are named after the algorithm they are made with.

To create the keyword-driven ontologies, keywords have to be extracted. Several keyword extraction methods exist. Instead of the keyword extraction method by Rospoucher et al. [3], which uses KX [50] to get an ordered list of keywords, we combine the Term Frequency (TF) and the term extraction method from Verberne et al. [51]. The standard Wikipedia corpus from the paper is used as background set. We combined the keywords of the two sets and manually deleted all subjectively determined non-relevant terms, resulting in the following set of 12 keywords for Agri: *Data, Food, Information, Drones, Agriculture, Crop, Technology, Agricultural, Production, Development, Farmers, Supply Chain*. And for the pizza case we use the following 13 keywords: *cheese, pizza, sauce, peppers, chicken, mozzarella, onion, tomato, pepperoni, mushroom, bacon, olive, italian* These keywords are used to create the Word2vec, WordNet and ConceptNet ontologies as well as the combined ontology of these three.

A. Hearst

Hearst patterns [52] can be used to extract hyponym relations, represented in an ontology as a 'IsA' relation. An example is 'Vegetable' is a hyponym of 'Food'. In unstructured texts, hyponyms can be spotted using the lexical structures 'NP, such as NP', or 'NP, or other NP', where NP is a noun phrase. These patterns are used to create an ontology with 'IsA' relations.

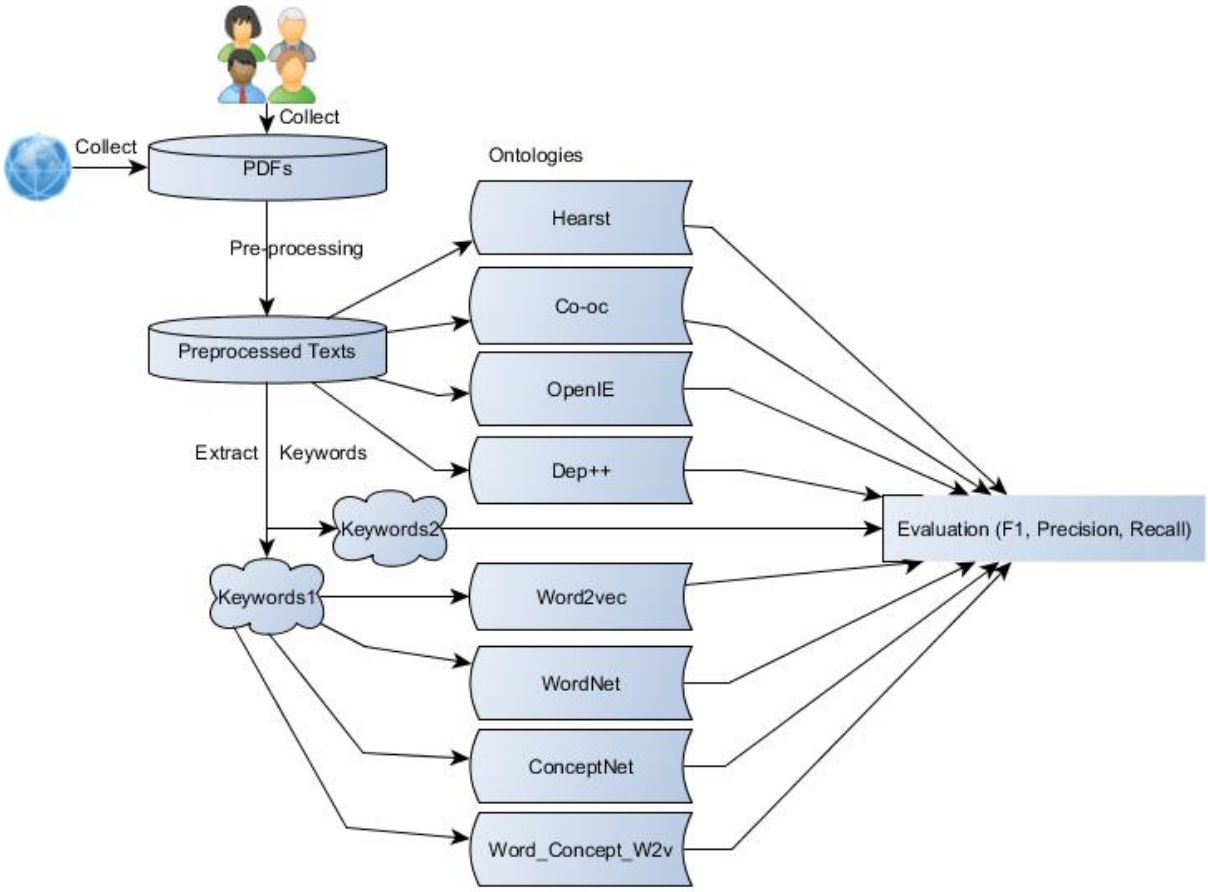

Figure 1. Overview of the methods to create the ontologies

B. Co-oc

Co-occurrences can extract all type of relations, because the number of times words co-occur with each other, for example in the same sentence, are counted [53]. We calculated the set of pairs of different words that co-occur in the sentences of the document set with a maximum distance of 4 words. Therefore, the N-gram generator of the CountVectorizer module of the Scikit-learn package [54] is used and the set is cleaned with the built-in English stopword list. As this set of co-occurring pairs of words will be very large, we further pruned the set using a threshold on the minimum number of times a pair of words co-occurs. This threshold is defined as a percentage of the maximum number of times a co-occurring pair of words is found. In the experiments, this number is set to 10. This number is based on experimentation with several values (ranging from 1 to 50) and overall performance seems best with 10 in our case. The ontology based on these co-occurring pairs of words will have only one vague 'co-occurrence' relation, indicating that the words that co-occur with each other in the document set. The specific type of relation is not determined.

C. OpenIE

The Open Information Extraction tool (OpenIE) is created by the CoreNLP group of Stanford [55]. The tools from the Stanford CoreNLP group are one of the most used tools in the NLP field. The OpenIE tool provides the whole processing chain from plain text through syntactic analysis (sentence splitter, part-of-speech tagger, dependency parser) to triples (object - relation - subject). The extracted relations are often the verbs in the sentence, and this results in triples, multiple word concepts, and many different relations.

D. Dep++

Similar to OntoCMaps [30], syntactic patterns are used to enhance the the Stanford Dependency Parser [56]. The algorithm consists of the following steps: a. Take each document in the corpus and generate sentences based on NLTK tokenization; b. Consider only sentences with more than 5 words which pass through the English check of the Python langdetect package; c. Parse each sentence through the Stanford DepParse annotator to generate Enhanced++Dependencies; d. Replace every word in the Enh++Dep by its lemma as produced by the Stanford POS tagger to consider only singular words; e. Then, generate a graph with a triple <governor,dependency,dependent> for each enhanced++dependency and apply the following transformation rules to the it:

- 1: Transform compound dependencies into 2-word concepts using rule: if $(X, compound, Y)$ then replace X with YX and remove Y
- 2: Enhance subject-object relations based on conjunction dependencies using rule: if $(X, nsubj, Y)$

and $(X, dobj, Z)$ and $(X, conj\_and, X')$ then add $(X', nsubj, Y)$ and $(X', dobj, Z)$

Finally, apply language patterns to derive triples from the dependency graph:

- pattern 1: if $(X, amod, Y)$ then add triple $(YX, subClassOf, X)$
- pattern 2: if $(X, compound, Y)$ and $(X \text{is} NN \text{or} NNS)$ then add triple $(YX, subClassOf, X)$
- pattern 3: if $(X, nsubj, Y)$ and $(X, dobj, Z)$ then add triple (Y, X, Z)

This algorithm yields an ontology that is similar to the OpenIE ontology, but should have less triples and thus less noise in it in terms of NLP-based constructs.

E. Word2vec

The first keyword-based method explained here is Word2vec. Word2vec is a group of models, which produce semantic embeddings. These models create neural word embeddings using a shallow neural network that is trained on a large dataset, such as Wikipedia, Google News or Twitter. Each word vector is trained to maximize the log probability of neighboring words, resulting in a good performance in associations, such as *king - man + woman = queen*. We use the skip-gram model with negative sampling (SGNS) [57] to create a semantic embedding of our agriculture documents. With the keywords, we search for the top ten most similar words and add a 'RelatedTo' relation between the keyword and this most similar word. This process is repeated for all most similar words.

F. WordNet

WordNet is a hierarchical dictionary containing lexical relations between words, such as synonyms, hyponyms, hypernyms and antonyms [58]. It also provides all possible meanings of the word, which are called *synsets*, together with a short definition and usage examples. WordNet contains over 155,000 words and over 206,900 word-sense pairs. We use the keywords to search in WordNet. We select the first synset (the most common), extract the 'Synonym' and 'Antonym' relations, and use these to create our ontology.

G. ConceptNet

ConceptNet (version 5) is a knowledge representation project in which a semantic graph with general human knowledge is build [59]. This general human knowledge is collected using other knowledge bases, such as Wikipedia and WordNet, and experts and volunteers. Some of the relations in ConceptNet are *RelatedTo*, *IsA*, *partOf*, *HasA*, *UsedFor*, *CapableOf*, *AtLocation*, *Causes*, *HasSubEvent*, *CreatedBy*, *Synonym* and *DefinedAs*. The strength of the relation is determined by the amount and reliability of the sources asserting the fact. Currently, ConceptNet contains concepts from 77 language and more than 28 million links between concepts. We use the keywords to search (through the API) in ConceptNet and extract all direct relations to create the ontology.

H. Word_Concept_W2v

This method takes the union (all relations) from the keyword-based methods WordNet, ConceptNet and Word2vec. This is, thus, a self created algorithm that combines the other three algorithms.

IV. Experimental Set-up

A. Document Sets

In our experiment, three different document sets are used, of which two are dedicated to pizzas and one is focused on our application domain of Agriculture. The document sets are described below.

a) PizzaMenus: The first pizza document set is the one created by Rospocher et al. [3]. This document set consists of 50 online available pizza restaurant menus, together approximately 22,000 words. The pizza types, ingredients, types of crust and details on sizes and prices are described, but also other information about beverages and other types of food such as sandwiches.

b) PizzaWiki: The other pizza document set is based on the information on Wikipedia. The original description of pizza is used, as well as all descriptions of pizza varieties and cooking varieties that were present as a box in the pizza description (as of date July 4th, 2019). This resulted in a set of 45 documents about pizza.

c) Agriculture: Our experts collected 135 articles on the Agriculture domain, including Agrifood, Agro-ecology, crop production and the food supply chain.

B. Ontologies

The methods used to create the ontologies are explained in the previous section. The keywords for the pizza domain are not dependent on the pizza document set, because they are partly manually created, so the ontologies for both the PizzaMenus and PizzaWiki is the same. Table I shows the number of classes in the ontologies and some examples of the relations with the word 'pizza' in both the PizzaMenus and PizzaWiki docset. Table II shows similar information for the Agriculture document set, with the word 'Agriculture'. These are randomly picked words from the extracted triples, just to give an idea of some of the words extracted by each of the methods.

TABLE I. INSIGHTS IN THE PIZZA ONTOLOGIES.

OntologyName	#Classes	#Relations	RelationPizza
Hearst_{menus}	4	2	tomato sauce specialty
Co-oc_{menus}	99	369	crust, bbq, onions, fresh
OpenIE_{menus}	840	887	dominos, sesame seeds, beef, anything
Dep++_{menus}	1218	916	dough, topping, sauce, you
Hearst_{wiki}	153	110	pizza chains, hawaiian pineapple, prezzo
Co-oc_{wiki}	113	164	crust, italian, topping, city
OpenIE_{wiki}	5690	8160	baked, popular, see food, topped
Dep++_{wiki}	3458	2953	frozen, crust, deep-fried, cheese
Word2vec	46	274	garlic, sauce, pepper, tomato
WordNet	54	53	pizza pie
ConceptNet	109	111	pepperoni, hamburger, deliver, oven
Word_Concept_W2v	171	438	oven, mushrooms, olives, green peppers

TABLE II. INSIGHTS IN THE SMARTGREEN ONTOLOGIES.

OntologyName	#Classes	#Relations	RelationAgriculture
Hearst	7523	7906	sector, yield forecasting, irrigation
Co-oc	1049	132,068	food, woman, adopt, production
OpenIE	280,063	535,380	sustainability, they, vision, water use
Dep++	178,338	205,251	sustainable, industrial, we, climate-smart
Word2vec	234	264	farming, biofuel, horticulture, innovation
WordNet	113	116	agribusiness, factory farm, farming
ConceptNet	203	213	farm, farmer, class, agribusiness
Word_Concept_W2v	491	593	agribusiness, farming, farm, horticulture

C. Evaluation

In order to evaluate the performance of the algorithms, it would be best to have a golden standard ontology for the domain that can be used as ground truth. Then, the challenge is to determine how close the created ontologies are to this golden standard ontology in terms of the number of concepts and relations in the created ontology that are also in the golden standard ontology.

Since we do not have a golden standard ontology in our agriculture case, a set of keywords is generated from the input document set using the KLdiv method and taken as ground truth. KLdiv is a proven good method for keyword extraction and therefore we assume that it generates keywords that are close to the ground truth with respect to the concepts that need to be present in the ontology. Then, the assumption is that the semantic quality of a created ontology is better if a keyword is present as concept in the ontology. It might be one of the best data-driven methods, but obviously not as good as human ground truth. The advantage is, though, that the number of keywords can be set. These evaluation-keywords are slightly different from our partly manually selected keyword set. For the pizza document sets, we calculate both the performance based on the keywords and the performance based on the pizza ontology [2] that is considered to be a golden standard ontology.

Although we cannot guarantee this is the best method to test the full range of the capabilities and performance of the algorithms, the three different datasets and the different number of keywords give a sense of the diversity in the results and the performance of the algorithms.

To evaluate the created ontologies, three different metrics to calculate a F1 score are used, which is based on a precision and a recall score. The first two metrics are based on the formulas proposed by Rospocher et al. [3] and the last metric also takes the relations between concepts into account.

D. Node-based F1

$$\begin{aligned} Prec_{node} &= \frac{k \in K_{correct}}{\#k \in Onto} \\ Rec_{node} &= \frac{k \in K_{correct}}{\#k \in K} \\ F1_{node} &= 2 * \frac{(Rec * Prec)}{Rec + Prec} \end{aligned} \tag{1}$$

where k is a keyword, which can be found in the set of correct keywords ($K_{correct}$), the total set of extracted keywords (K) and in the ontology ($Onto$) to be evaluated.

E. Weighted Node-based F1

$$\begin{aligned} Prec_{wnode} &= \frac{k \in K_{correct}}{\#k \in Onto} \\ Rec_{wnode} &= \frac{\sum(rel_{kcorrect}) \in K_{correct}}{\sum(rel_k) \in K} \\ rel_k = F1_{wnode} &= 2 * \frac{(Rec * Prec)}{Rec + Prec} \end{aligned} \tag{2}$$

where k is a keyword, which can be found in the set of correct keywords ($K_{correct}$), the total set of extracted keywords (K) and in the ontology ($Onto$) to be evaluated. $rel_kcorrect$ is the sum of the relevance scores in $K_{correct}$ and rel_k is the sum of the relevance scores in K. The relevance scores are determined through the KLdiv weights.

F. Relation-based F1

$$\begin{aligned} Prec_{rel} &= \frac{\#r \in R \text{ with } k \in K}{\#r \in R} \\ Rec_{rel} &= \frac{\#k \in K \text{ found in } R}{\#k \in K} \\ F1_{rel} &= 2 * \frac{(Rec * Prec)}{Rec + Prec} \end{aligned} \tag{3}$$

where k is keyword in set of Keywords (K), r is relation in set of Relations (R). The set of selected items is thus the set of relations R (precision), and the set of relevant items is thus the set of keywords K (recall).

V. Results

A. PizzaMenus and PizzaWiki

Figure 2 shows for each ontology the overall quality based on the F1 score of the keywords for 15, 30, 50, 100, 150 and 200 keywords, Figure 3 for the weighted version and Figure 4 for the relation-based version.

The results show that in the node based F1 methods word2vec seems to be the best method until 100 keywords. Co-oc is the second best. The pizza menus document set gives slightly better results compared to the PizzaWiki document set. In the relation-based evaluation, the combined keyword-based methods word_concept_w2v scores high, as well as the Co-oc Wiki. The Wiki document set seems slightly better compared to the Menus document set with this evaluation metric.

Table III shows the results of the node-based F1 scores based on the pizza ontology [2]. The concepts of the pizza ontology are used as keywords. The pizza ontology can contain multiple words in one concept, but we define a correct concept in the created ontology to be evaluated has to match at least one word in a golden keyword. The keyword weight is set to the number of edges of the keyword in the golden ontology.

The results of the table show that Co-oc of the Menus document set is the best method for the normal F1, whereas word2vec is the best with the weighted F1 metric. This is similar to the results with the keywords.

TABLE III. F1 SCORES OF THE PIZZA ONTOLOGIES BASED ON THE GROUND TRUTH

OntologyName	F1 node	F1 weighted node
Hearst$_{menus}$	0.019	0.022
Co-oc$_{menus}$	0.311	0.369
OpenIE$_{menus}$	0.043	0.045
Dep++$_{menus}$	0.051	0.052
Hearst$_{wiki}$	0.067	0.081
Co-oc$_{wiki}$	0.211	0.244
OpenIE$_{wiki}$	0.013	0.013
Dep++$_{wiki}$	0.022	0.022
Word2vec	0.283	0.403
WordNet	0.164	0.260
ConceptNet	0.190	0.239
Word_Concept_W2v	0.177	0.203

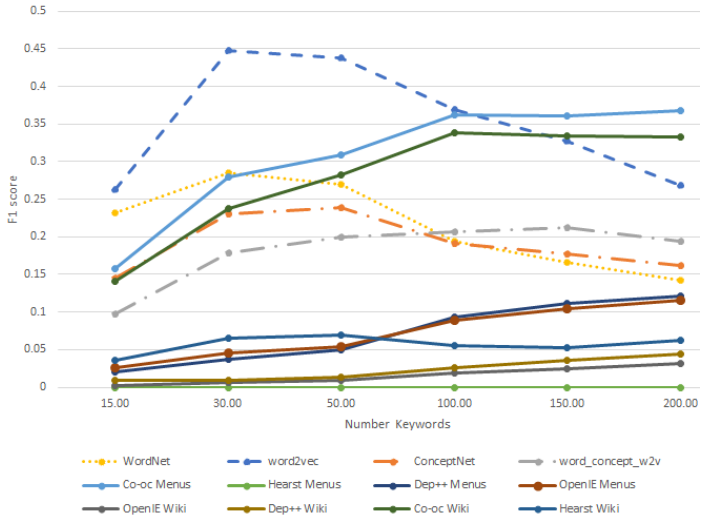

Figure 2. Node-based F1 score for Pizza datasets

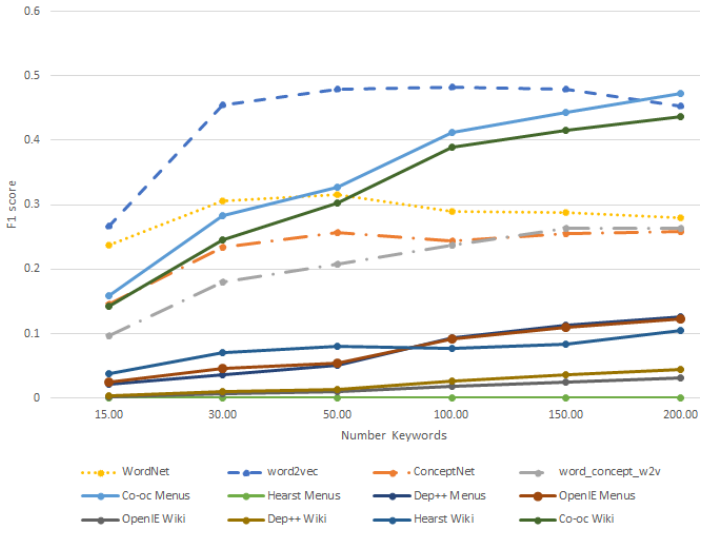

Figure 3. Weighted Node-based F1 score for Pizza datasets

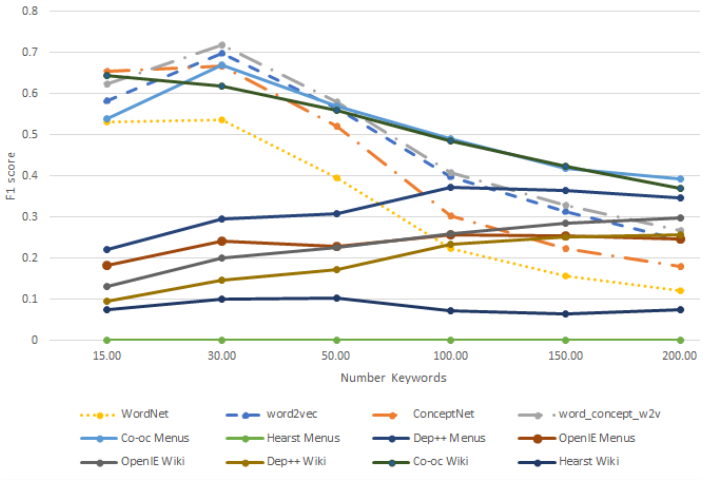

Figure 4. Relation-based F1 score for Pizza datasets

B. Agriculture

Figure 5 shows for each ontology the overall quality based on the F1 score of the keywords for 15, 30, 50, 100, 150 and 200 keywords, Figure 6 for the weighted version and Figure 7 for the relation-based version.

The results show that in the node-based F1 scores, the single keyword-based methods, especially WordNet, are better than the document-based methods. After 50 keywords, the Co-oc method is the highest scoring method and performance becomes twice as good compared to the other methods. The difference between the normal node-based method and the weighted method is mainly visible in the WordNet score, which declines less in the weighted version. The trend of the lines in the relation-based evaluation is different from the node-based evaluation methods. Co-oc is scores highest for all number of keywords. The keyword-based methods perform better compared to the node-based evaluation and with more keywords even outperform some of the keyword-based methods. The combined word_concept_w2v method is in the relation based evaluation better compared to the single methods.

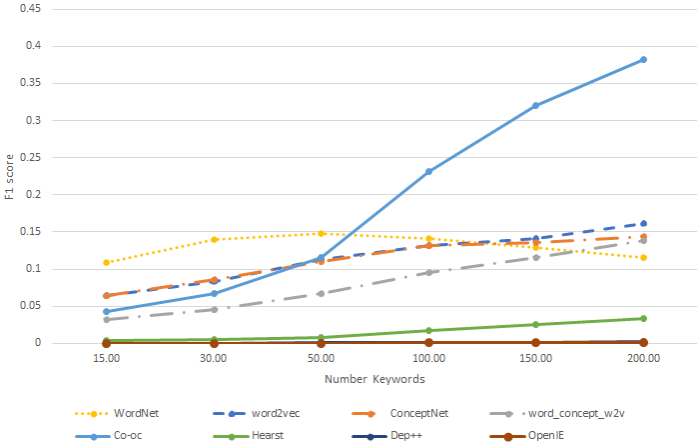

Figure 5. Node-based F1 score for SmartGreen dataset

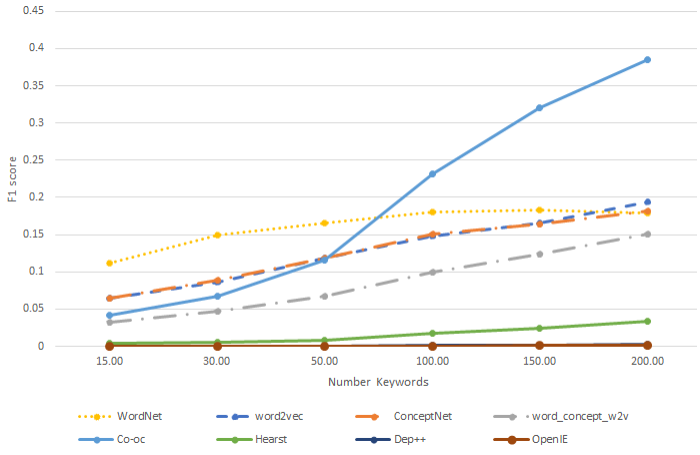

Figure 6. Weighted Node-based F1 score for SmartGreen dataset

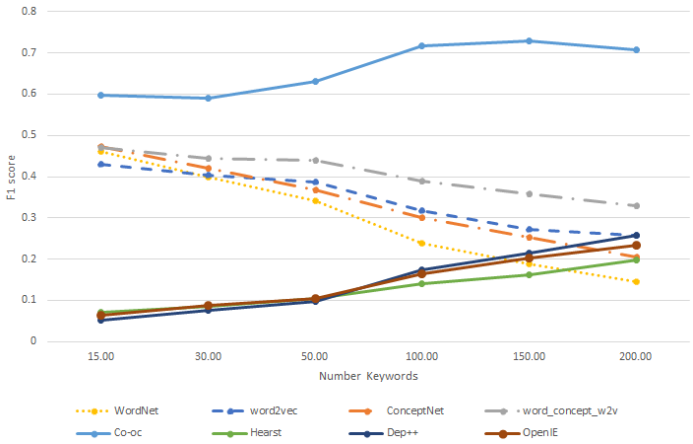

Figure 7. Relation-based F1 score for SmartGreen dataset

VI. Discussion

In this section, we discuss the results of our experiments by comparing the document sets, algorithms and evaluation methods with each other.

First, when we compare the results between the PizzaMenus and PizzaWiki docsets, we see that only Hearst is better for PizzaWiki, because only 2 matches were found in Menus. For all other algorithms PizzaMenus is better compared to PizzaWiki. For example in Co-oc, 28 matches are found in PizzaMenus, whereas 17 are found in PizzaWiki. We expected this to be the other way around, because usually Wikipedia descriptions contain better natural language phrases than menus. We conclude therefore, that the menus do not contain a lot of correct English sentences, but do contain the correct keywords. Overall, Word2Vec is best when 100 evaluation-keywords or less are used. Above 100, Co-oc becomes best and remains at the same level with increased number of evaluation-keywords. Another interesting result is that WordNet is not better then Word2Vec. A reason can be that WordNet adds a few synonyms to the ontology that are not keywords and thus decreases precision and F1.

Second, when we compare the different evaluation metrics, we see that for both the Pizza and Agriculture docsets, the curves for node-based and weighted node-based F1 scores are very similar. In absolute terms, the weighted node-based F1 score is slightly better. Comparing our results of the PizzaMenus docset to the results of Rospocher et al. [3], they report an F1 score of 0.17 and a weighted F1 of 0.25 on the PizzaMenus. They enrich the keywords that are extracted with WordNet concepts, which is similar to our WordNet algorithm. Fortunately, we can see that our F1 score of 0.18 and weighted F1 score of 0.26 is almost equal. On the other hand, the curves of our relation-based F1 score look different. When the number of evaluation-keywords increases the F1-score decreases for most of the algorithms, except for Co-oc in the Agriculture docset. This is due to a stronger decrease of the recall with increasing number of evaluation-keywords. An explanation for this effect might be that the counters in the relation-based precision and recall definitions differ in contrast to the node-based precision and recall definitions.

Third, when we compare the F1 scores of the Pizza document sets with the Agriculture docset, we see that in the Pizza document sets both Word2Vec and Co-oc perform best, while in the Agriculture docset Co-oc outperforms the other algorithms. Specifically, Co-oc becomes best with increased number of evaluation-keywords above 60, which is lower than the 100 evaluation-keywords with the Pizza docsets. So, the main difference between the results of the Pizza docsets and the Agriculture docset is the curve of the Word2Vec algorithm. This can be explained as follows. In principle, the F1 score of every algorithm follows a parabolic form with a peak. The peaks of the curves of the keyword-based algorithms appear in the range of 15 to 200 evaluation-keywords for the Pizza docsets but not for the Agriculture docset. This is dependent on the number of concepts in the generated ontologies and thus on the size of the docset. In general, we can conclude that the peak in the F1-score appears when the number of evaluation-keywords is close to the number of concepts in the ontology that is evaluated.

Fourth, when looking at the purely NLP-based algorithms OpenIE and Dep++, we see that both do not perform very well. The number of concepts and relations generated by Dep++ are considerably less compared to those generated by OpenIE. The number of matching concepts is approximately the same for both algorithms. Therefore, we conclude that the Dep++ algorithm generates a smaller and slightly better ontology, but that this ontology still contains a lot of 'noise' in terms of non-relevant concepts and relations.

Finally, we compare keyword-based algorithms with the document based algorithms. In general the keyword-based algorithms are better than the document-based ones. The only exception is the Co-oc algorithm that outperforms all other algorithms. An explanation of this effect can be that apparently the main keywords of the docset often appear close to each other and are, therefore, part of the Co-oc generated ontology. Despite that fact, the keyword-based methods have the advantage that 1) they are based on general knowledge bases and thus no domain-specific documents have to be collected and 2) in an expert session to build up a domain ontology only a set of keywords have to be generated and agreed upon. A disadvantage is that when a topic becomes very specific, common knowledge bases have very sparse information, whereas domain-specific documents might provide more information. This is also visible in the number of classes and relations found for the different methods. We advise to start with building a domain ontology based on generic topics that are present in the domain using the keyword-based algorithms, preferably Word2Vec. When domain-specific topics need to be added, it is better to use the Co-oc algorithm based on a docset of specific domain documents.

We conclude this discussion with the following issues:

- For the pizza domain, the use of a set of menu documents is feasible, but for most domains this is not possible and it is, therefore, better to use Wikipedia as a basis for finding documents or articles about the domain.
- The current definitions of precision and recall are mostly based on concepts and partly on relations. Thus, a better definition that fully takes relations account is needed. This is one of our future work items.
- With the current results and precision and recall definitions we can achieve F1 scores of up to around 0.7.

Using more strict definitions of precision and recall, this will most probably drop down below 0.5. The question is whether this performance results in domain ontologies that are acceptable for domain experts to function as a head start for an expert meeting. Future experiments in which domain experts are involved in the evaluation phase are therefore necessary.

VII. Conclusion

Creating ontologies is takes time and effort. In this paper, we examined whether we can create data-driven ontologies based on a set of documents to start an ontology-creation session with a head start. In our experiments, we compare 8 different data-driven algorithms, four based on the documents themselves and four based on extracted keywords. We use two pizza document sets and one agriculture document set to generate ontologies with these algorithms. Finally, we use three different evaluation metrics to compare performance in terms of precision, recall and F1-score.

The results show that the keyword-based methods in general outperform the document-based methods. The only exception is Co-oc. Based on these results, we suggest to use the Word2Vec method in a domain with general topics and shift to Co-oc for specific topics for which no information is present in common knowledge bases. The different evaluation metrics show similar trends. The advantage of the relation-based metric is that the relation is taken into account. Future work should be done to further optimize and validate this metric in order to define correct relations of the ontologies. Another topic of future work is to qualitatively validate the ontologies, for instance using the layered ontology metrics suite for ontology assessment with syntactic, semantic, pragmatic and social quality criteria. Using human evaluators it can be verified whether the current performance is high enough to be valuable to use as a head start in an ontology creation session.

Acknowledgment

This work has been executed as part of the Interreg Smart-Green project (https://northsearegion.eu/smartgreen/). The authors would like to thank Christopher Brewster for providing a representative agriculture-related document set and Roos Bakker for proof-reading our paper.

References

[1] M. de Boer and J. Verhoosel, "Creating data-driven ontologies: An agriculture use case," in ALLDATA 2019: the Fifth International Conference on Big Data, Small Data, Linked Data and Open Data, Valencia, Spain 24-28 March 2019, 52-57, 2019.

[2] "Pizza.owl," https://protege.stanford.edu/ontologies/pizza/pizza.owl, accessed: 2019-07-25.

[3] M. Rospocher, S. Tonelli, L. Serafini, and E. Pianta, "Corpus-based terminological evaluation of ontologies," Applied Ontology, vol. 7, no. 4, 2012, pp. 429–448.

[4] P. Cimiano, A. Mädche, S. Staab, and J. Völker, "Ontology learning," in Handbook on ontologies. Springer, 2009, pp. 245–267.

[5] C. A. Brewster, "Mind the gap: Bridging from text to ontological knowledge," Ph.D. dissertation, University of Sheffield, 2008.

[6] S. Gillani and A. Kő, "Promine: a text mining solution for concept extraction and filtering," in Corporate Knowledge Discovery and Organizational Learning. Springer, 2016, pp. 59–82.

[7] J. Park, W. Cho, and S. Rho, "Evaluating ontology extraction tools using a comprehensive evaluation framework," Data & Knowledge Engineering, vol. 69, no. 10, 2010, pp. 1043–1061.

[8] R. Glauber and D. B. Claro, "A systematic mapping study on open information extraction," Expert Systems with Applications, vol. 112, 2018, pp. 372–387.

[9] C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh, "A survey on open information extraction," arXiv preprint arXiv:1806.05599, 2018.

[10] A. Yates and et al., "Textrunner: open information extraction on the web," in Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. Association for Computational Linguistics, 2007, pp. 25–26.

[11] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010, pp. 118–127.

[12] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2011, pp. 1535–1545.

[13] M. Schmitz, R. Bart, S. Soderland, O. Etzioni et al., "Open language learning for information extraction," in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012, pp. 523–534.

[14] L. Del Corro and R. Gemulla, "Clausie: clause-based open information extraction," in Proceedings of the 22nd international conference on World Wide Web. ACM, 2013, pp. 355–366.

[15] L. Cui, F. Wei, and M. Zhou, "Neural open information extraction," arXiv preprint arXiv:1805.04270, 2018.

[16] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2016, pp. 2124–2133.

[17] S. Kumar, "A survey of deep learning methods for relation extraction," arXiv preprint arXiv:1705.03645, 2017.

[18] R. Alfred and et al., "Ontology-based query expansion for supporting information retrieval in agriculture," in The 8th International Conference on Knowledge Management in Organizations. Springer, 2014, pp. 299–311.

[19] M. Song, I.-Y. Song, X. Hu, and R. B. Allen, "Integration of association rules and ontologies for semantic query expansion," Data & Knowledge Engineering, vol. 63, no. 1, 2007, pp. 63–75.

[20] M. de Boer, K. Schutte, and W. Kraaij, "Knowledge based query expansion in complex multimedia event detection," Multimedia Tools and Applications, vol. 75, no. 15, 2016, pp. 9025–9043.

[21] M. H. de Boer and et al., "Query interpretation–an application of semiotics in image retrieval," International Journal on Advances in Software, vol. 3 4, 2015, pp. 435–449.

[22] M. H. De Boer, Y.-J. Lu, H. Zhang, K. Schutte, C.-W. Ngo, and W. Kraaij, "Semantic reasoning in zero example video event retrieval," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 13, no. 4, 2017, p. 60.

[23] İ. Pembeci, "Using word embeddings for ontology enrichment," International Journal of Intelligent Systems and Applications in Engineering, vol. 4, no. 3, 2016, pp. 49–56.

[24] G. Wohlgenannt and F. Minic, "Using word2vec to build a simple ontology learning system," in International Semantic Web Conference (Posters & Demos), 2016.

[25] B. Biebow, S. Szulman, and A. J. Clément, "Terminae: A linguistics-based tool for the building of a domain ontology," in Int. Conf. on Knowledge Engineering and Knowledge Management. Springer, 1999, pp. 49–66.

[26] P. Buitelaar, D. Olejnik, and M. Sintek, "A protégé plug-in for ontology extraction from text based on linguistic analysis," in European Semantic Web Symposium. Springer, 2004, pp. 31–44.

[27] T. Declerck, "A set of tools for integrating linguistic and non-linguistic information," in Proceedings of SAAKM (ECAI Workshop), 2002.

[28] P. Cimiano and J. Völker, "text2onto," in Int. Conf. on Appl. of Nat. Lang. to Inf. Sys. Springer, 2005, pp. 227–238.

[29] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: an architecture for development of robust hlt applications," in Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002, pp. 168–175.

[30] A. Zouaq, "An overview of shallow and deep natural language processing for ontology learning," in Ontology learning and knowledge discovery using the web: Challenges and recent advances. IGI Global, 2011, pp. 16–37.

[31] X. Jiang and A.-H. Tan, "Crctol: A semantic-based domain ontology learning system," Journal of the American Society for Information Science and Technology, vol. 61, no. 1, 2010, pp. 150–168.

[32] Y.-B. Kang, P. D. Haghighi, and F. Burstein, "Cfinder: An intelligent key concept finder from text for ontology development," Expert Systems with Applications, vol. 41, no. 9, 2014, pp. 4494–4504.

[33] H. Poon and P. Domingos, "Unsupervised ontology induction from text," in Proceedings of the 48th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010, pp. 296–305.

[34] A. Gangemi, V. Presutti, D. Reforgiato Recupero, A. G. Nuzzolese, F. Draicchio, and M. Mongiovì, "Semantic web machine reading with fred," Semantic Web, vol. 8, no. 6, 2017, pp. 873–893.

[35] I. Tiddi, N. B. Mustapha, Y. Vanrompay, and M.-A. Aufaure, "Ontology learning from open linked data and web snippets," in OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, 2012, pp. 434–443.

[36] R. Bendaoud, A. M. R. Hacene, Y. Toussaint, B. Delecroix, and A. Napoli, "Text-based ontology construction using relational concept analysis," in International Workshop on Ontology Dynamics-IWOD 2007, 2007.

[37] S. Mittal, A. Joshi, T. Finin et al., "Thinking, fast and slow: Combining vector spaces and knowledge graphs," arXiv, no. arXiv: 1708.03310, 2017.

[38] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in European Semantic Web Conference. Springer, 2018, pp. 593–607.

[39] J. Brank, M. Grobelnik, and D. Mladenić, "A survey of ontology evaluation techniques," 2005.

[40] H. Hlomani and D. Stacey, "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey," Semantic Web Journal, vol. 1, no. 5, 2014, pp. 1–11.

[41] P. Spyns, "Evalexon: Assessing triples m ined from texts," STAR, vol. 9, 2005, p. 09.

[42] H. Hlomani and D. A. Stacey, "Contributing evidence to data-driven ontology evaluation workflow ontologies perspective," in 5th International Conference on Knowledge Engineering and Ontology Development, KEOD 2013, 2013, pp. 207–213.

[43] L. Ouyang, B. Zou, M. Qu, and C. Zhang, "A method of ontology evaluation based on coverage, cohesion and coupling," in Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on, vol. 4. IEEE, 2011, pp. 2451–2455.

[44] C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks, "Data driven ontology evaluation," 2004.

[45] M. Poveda-Villalón, A. Gómez-Pérez, and M. C. Suárez-Figueroa, "Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation," International Journal on Semantic Web and Information Systems (IJSWIS), vol. 10, no. 2, 2014, pp. 7–34.

[46] A. Burton-Jones, V. C. Storey, V. Sugumaran, and P. Ahluwalia, "A semiotic metrics suite for assessing the quality of ontologies," Data & Knowledge Engineering, vol. 55, no. 1, 2005, pp. 84–102.

[47] A. Gangemi and V. Presutti, "Ontology design patterns," in Handbook on ontologies. Springer, 2009, pp. 221–243.

[48] A. Lozano-Tello and A. Gómez-Pérez, "Ontometric: A method to choose the appropriate ontology," Journal of Database Management (JDM), vol. 15, no. 2, 2004, pp. 1–18.

[49] M. McDaniel, V. C. Storey, and V. Sugumaran, "Assessing the quality of domain ontologies: Metrics and an automated ranking system," Data & Knowledge Engineering, vol. 115, 2018, pp. 32–47.

[50] E. Pianta and S. Tonelli, "Kx: A flexible system for keyphrase extraction," in Proceedings of the 5th international workshop on semantic evaluation. Association for Computational Linguistics, 2010, pp. 170–173.

[51] S. Verberne, M. Sappelli, D. Hiemstra, and W. Kraaij, "Evaluation and analysis of term scoring methods for term extraction," Information Retrieval Journal, vol. 19, no. 5, 2016, pp. 510–545.

[52] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in Proceedings of the 14th conference on Computational linguistics-Volume 2. Association for Computational Linguistics, 1992, pp. 539–545.

[53] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," International Journal on Artificial Intelligence Tools, vol. 13, no. 01, 2004, pp. 157–169.

[54] "Countvectorizer," https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html, accessed: 2019-07-25.

[55] G. Angeli, M. J. J. Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in Proc. of 53 ACL and 7th Int. Joint Conf. on NLP (Vol 1: Long Papers), vol. 1, 2015, pp. 344–354.

[56] M.-C. De Marneffe, B. MacCartney, and C. D. Manning, "Generating typed dependency parses from phrase structure parses," 2006.

[57] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Adv. in neural information processing systems, 2013, pp. 3111–3119.

[58] G. A. Miller, "Wordnet: a lexical database for english," Communications of the ACM, vol. 38, no. 11, 1995, pp. 39–41.

[59] R. Speer and C. Havasi, "Representing general relational knowledge in conceptnet 5." in LREC, 2012, pp. 3679–3686.