Creating Data-Driven Ontologies
An Agriculture Use Case

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

Data Science department, TNO,
Anna van Buerenplein 1, 2595 DA, The Hague, The Netherlands
Email: maaike.deboer@tno.nl

Abstract—**The manual creation of an ontology is a tedious task. In the field of ontology learning, Natural Language Processing (NLP) techniques are used to automatically create ontologies. In this paper, we present a methodology using data-driven techniques to create ontologies from unstructured documents in the agriculture domain. We use state-of-the-art NLP techniques based on Stanford OpenIE, Hearst patterns and co-occurrences to create ontologies. We added an NLP-method that uses dependency parsing and transformation rules based on linguistic patterns. In addition, we use keyword-driven techniques from the query expansion field, based on Word2vec, WordNet and ConceptNet, to create ontologies. We added a method that takes the union of the ontologies produced by the keyword-based methods. The semantic quality of the different ontologies is calculated using automatically extracted keywords. We define recall, precision and F1-score based on the concepts and relations in which the keywords are present. The results show that 1) the method based on co-occurrences has the best F1-score with more than 100 keywords; 2) the keyword-based methods have a higher F1-score than the NLP-based methods with less than 100 keywords in the evaluation and; 3) the combined keyword-based method always has a higher F1-score compared to each single method. In our future work, we will focus on improving the dependency parsing algorithm, improving combining different ontologies, and improving our quality evaluation methodology.**

Keywords–Knowledge engineering; Machine Learning; Agriculture

I. INTRODUCTION

In the previous decade, data scientists often use 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. Advantages of this type of approach is that it is insightful, validated by experts, and it gives a feeling of control. 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 (only possible in closed-world domains) and that there might not be one truth [1]. If two experts create a knowledge model, they probably will come up with different ones, because each expert has his own subjective view of important concepts and relations in the domain. On the other hand, 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 is that the models are often not insightful, 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. A field in which ontologies learn from available knowledge using data is named *ontology learning*. We present in this paper an ontology learning methodology that uses existing and new data-driven algorithms to create ontologies based on unstructured textual documents in the agriculture domain. This results in an initial ontology that serves as a good starting point for further improvement by experts in the domain. The goal of this methodology is to create an improved ontology that can be used for semantic interoperability between IT-systems and human users. We use state-of-the-art techniques in ontology learning to create ontologies. Additionally, we use keyword-based techniques to create ontologies. These keywords are used to find relations in external knowledge bases and a word embedding model.

In order to evaluate the performance of our methodology, we measure the semantic quality of the resulting ontologies using a keyword-based method. We define recall, precision and F1-score based on automatically extracted keywords and their appearance in the ontologies. From a semantic point of view, the extracted ontology should therefore be evaluated on the number of important keywords it contains. From a usability point of view, it is important that the ontology is still comprehensible for the human user. We use a well-known keyword extraction algorithm to get the main keywords from the document set and limit the amount of evaluation keywords in the ontology.

In the next section, we describe the related work on ontology learning and the evaluation of ontologies. In Section 3, we explain the methods used to create the different ontologies. Section 4 describes our evaluation methodology and presents our results and Section 5 contains a discussion and conclusion as well as a description of future work.

II. RELATED WORK

A. Ontology Learning

Ontology learning is focused on learning ontologies based on data [2] [3]. One of the most known concepts in ontology learning is the ontology learning layer cake. Starting from the bottom of the cake, the order is terms, synonyms, concept formation, concept hierarchy, relations, relation hierarchy, axiom schemata and finally general axioms. Similar to the layed cake, Gillani et al. [4] 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 manners: structured, semi-structured and unstructured data [2]. Besides the manner of learning, there are three types of tools available: ontology editing tools, ontology merging tools and ontology extraction tools [5]. In this paper, we want to automatically create ontologies from text, so we focus on unstructured data and ontology extraction tools. Several tools are already available.

Some tools only focus on the information extraction, up to the relation extraction part. This subfield is also named Open Information Extraction (OpenIE). One of the first in this category is TextRunner [6]. TextRunner tags sentences with part-of-speech tags and noun phrase chunks, in a fast manner with one loop over all documents. The Resolver system does unsupervised clustering of the extractions to create sets of synonymous entities and relations. TextRunner was followed by WOE, ReVerb, KrakeN, EXEMPLAR, OLLIE, PredPatt, ClausIE, OpenIE4, CSD-IE, NESTIE, MinIE and Graphene [7]. Recently, deep learning methods, such as the encoder-decoder framework from Cui et al. [8] are proposed.

Related to the OpenIE field, query expansion can also be used to find more concepts and relations [9]. This method is often used in the information retrieval field. The most common method is to use WordNet [10]. Boer et al. also [1] [11] use ConceptNet to find related concepts and their relations. Word2vec is also used in information retrieval [12], ontology enrichment [13] and ontology learning [14].

One of the oldest methods that use the full ontology learning layered cake seems to be Terminae [15]. Terminae is a method and platform for ontology engineering, and includes linguistic analysis with 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.

OntoLT [16] 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) [17], which provides annotation of part-of-speech, 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.

Text2Onto [18] uses GATE to extract entities. GATE [19] has a submodule named ANNIE that contains a tokeniser, sentence splitter, Part-of-Speech (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 learned knowledge is stored at a meta-level in the form of instantiated modelling primitives. The model is, therefore, robust to different languages and changing information. According to Zouaq et al. [20], Text2Onto generates very shallow and light weight ontologies.

Concept-Relation-Concept Tuple based Ontology Learning (CRCTOL) [21] 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 relevant 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. [4], 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.

CFinder [22] 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 and a domain specific df as weight. The paper stops at the key concept extraction and does not go further with determining relations.

OntoUPS [23] 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. They use Markov Logical Networks (MLNs) for that.

OntoCMaps [20] uses the Stanford POS tagger and dependency parser to extract concepts. They use several generic patterns to extract relations.

Promine [4] use 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 concept are filtered using the information gain, path length and depth of concepts.

More recently, Mittal et al. [24] 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. [25] 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.

B. *Evaluating ontologies*

Brank et al. [26] state that most approaches to evaluate ontologies can be place 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. [27] also uses 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 good in one application does not have to be good 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, is not the case. Finally, the disadvantage of the human assessment is subjectivity.

In this paper, we focus on the data-driven evaluation. We do not have a golden standard, or an application, which leaves us with a data-driven or human assessment approach. 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 [28]–[31]. For example, Brewster et al.

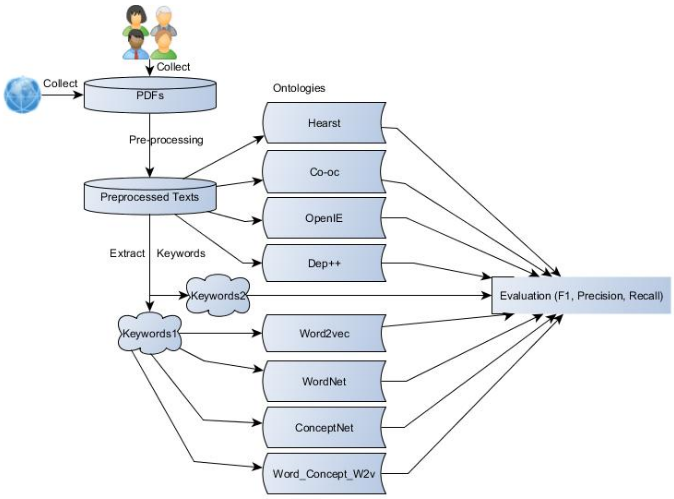

Figure 1. Overview of the methods to create the ontologies

Figure 2. Terms Extracted with the method from Verberne et al. [36]

[31] compare extracted terms and relations from text with the concepts and relations in the ontology. They use a probabilistic model to determine the best ontology for a certain domain.

Besides the categories, ontologies can be evaluated on different levels. These levels are defined differently in different papers. Brank et al. [26] divides the levels in lexical, hierarchical, other semantic relations, context, syntactic, 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. [32] is syntactic, semantic, pragmatic and social. Gangemi et al. [33] use the distinction between structural, functional and usability-profiling. Burton et al. [32] use lawfulness, richness, interpretability, consistency, clarity, comprehensiveness, accuracy, relevance, authority, and history. Lozano et al. [34] even use a three-level framework of 117 criteria. Hlomani et al. [27] 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. Recently, Mcdaniel et al. [35] 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). Figure 1 shows an overview of the methods used to create the ontologies. Our experts collected 135 articles on the Agriculture domain, including Agrifood, Agro-ecology, crop production and the food supply chain. From each article we first extracted the plain text from the PDF. On these plain texts we used sentence splitting, tokenizing, removing non-ascii and non-textual items as pre-processing. With these pre-processed texts we created the state-of-the-art ontologies Hearst, Co-oc and OpenIE (explained below). We also added a new method based on Dependencies and some rules.

To create the keyword-driven ontologies, we needed to extract keywords. We used the Term Frequency (TF) and the term extraction method from Verberne et al. [36]. The result of this term extraction method is shown in Figure 2. The standard Wikipedia corpus from the paper is used as background set. We combined the keywords of the two sets and manually deleted all non-relevant terms, resulting in the following set of keywords: *Data, Food, Information, Drones, Agriculture, Crop, Technology, Agricultural, Production, Development, Farmers, Supply Chain*. These keywords were used to create the Word2vec, WordNet and ConceptNet ontologies.

a) Hearst: Hearst patterns [37] 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.

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 [38]. We used a maximum distance of four words to calculate the co-occurrences. The ontology based on co-occurrences, thus, will have many classes and one vague relation, i.e. that the classes have co-occurred with each other in documents.

c) OpenIE: The Open Information Extraction tool (OpenIE) is created by the CoreNLP group of Stanford [39]. The tools from the Stanford CoreNLP group are one of the most used tools in the NLP field. The OpenIE tool provides the whole 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 a lot of triples multiple word concepts and a lot of different relations.

d) Dep++: Similar to OntoCMaps [20], we use patterns to enhance the the Stanford Dependency Parser [40]. The algorithm consists of the following steps. Take each document in the corpus and generate sentences based on NLTK tokenization. Consider only sentences with more than 5 words which pass through the English check of the Python langdetect package. Parse each sentence through the Stanford DepParse annotator to generate Enhanced++Dependencies. Replace every word in the Enh++Dep by its lemma as produced by the Stanford POS tagger to consider only singular words. 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 noise in it in terms of NLP-based constructs.

e) 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 huge 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) [41] 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 [42]. 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) and extract the 'Synonym' and 'Antonym' relations and use these to create our ontology.

g) ConceptNet: ConceptNet (5) is a knowledge representation project in which a semantic graph with general human knowledge is build [43]. 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.

IV. Results

To evaluate the created ontologies we use a simple but effective keyword-based evaluation method based on the framework of Brewster et al. [31]. Our evaluation algorithm first generates a set of keywords K from the document set using the *KLDiv* term extraction algorithm [36]. Then, the assumption is that the semantic quality of an ontology is better if a keyword is present as concept in a relation in the ontology. If we define an ontology as being a set of relations R between concepts, then we can defined keyword-based recall, precision and F1-score as follows:

$$\begin{aligned} Prec &= \frac{\#r \in R \text{ with } k \in K}{\#r \in R} \\ Rec &= \frac{\#k \in K \text{ found in } R}{\#k \in K} \\ F1 &= 2 * \frac{(Rec * Prec)}{Rec + Prec} \end{aligned} \quad (1)$$

, 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).

Figure 3 shows for each ontology the overall quality based on the F1 score for 15, 30, 50, 100, 150 and 200 keywords, Figure 4 and Figure 5 show the precison and recall. Table I shows the number of classes and some example of the relations with the word 'Agriculture'.

TABLE I. INSIGHTS IN THE DIFFERENT ONTOLOGIES.

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

The results show that the combined keyword-based methods (light-blue line) are always better than any of the three separate methods (WordNet, ConceptNet and Word2vec). The keyword-based methods have a higher F1 score with a lower number of keywords, whereas the NLP-based methods have a higher F1 score with a higher number of keywords. With 200 keywords, the best performing method is Co-oc, the method based on co-occurrences.

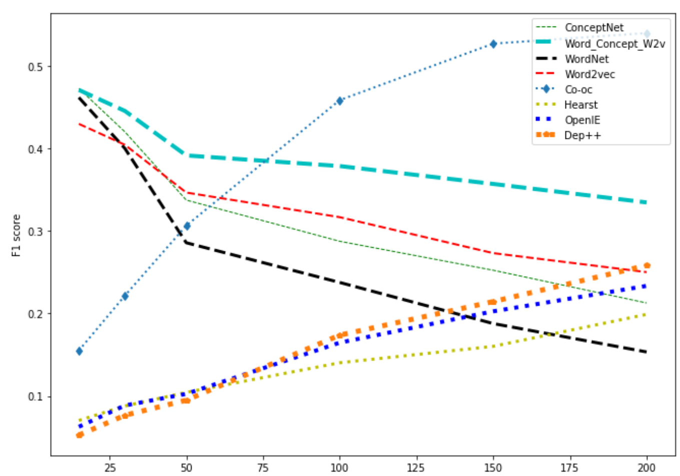

Figure 3. F1 score for the different methods

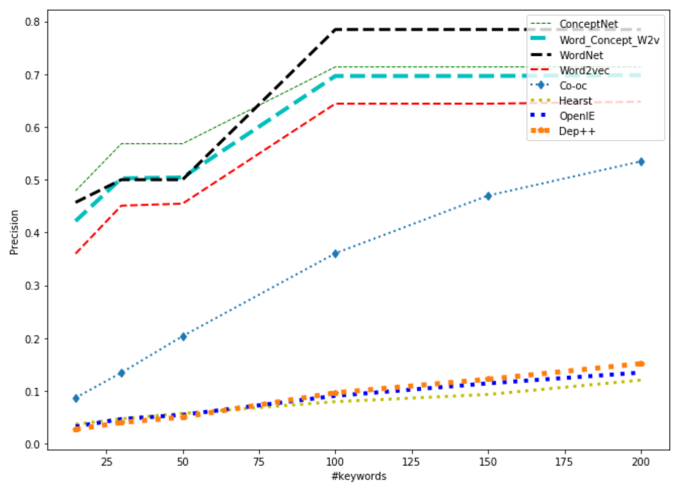

Figure 4. Precision score for the different methods

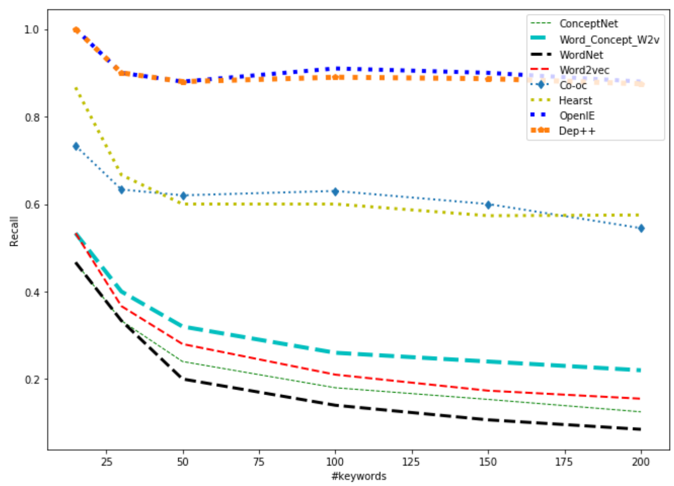

Figure 5. Recall score for the different methods

V. Discussion, Conclusion and Future Work

In this paper, we presented a methodology to use existing and new algorithms to create data-driven ontologies based on unstructured textual documents in the agriculture domain. In addition, we used a data-driven method based on keywords to evaluate the semantic quality of the ontologies. The goal of this methodology is to generate an initial ontology that serves as a good starting point for further improvement by experts in the domain. The resulting improved ontology can then be used for semantic interoperability between IT-systems and human users.

The results show that the keyword-based methods have the highest F1-score for less than 100 keywords. This can be validated by looking at the number of concepts. The keyword-based methods have less concepts than the NLP-based methods. If one keyword is found by the keyword-based methods, the precision will be already much higher than the precision of the NLP-based methods. When the number of evaluation-keywords increases the keyword-based methods perform less, because they contain less concepts. One critical note is that we based the keywords-based methods and the evaluation method on the same set of documents. Although we used a manual selection on the keywords and multiple keyword-based methods to find the set of keywords, we know that there might have been some overlap in the keywords used in creating the ontologies and the evaluation. This also clarifies why the performance on a few keywords is higher for the keyword-based methods. The NLP-based methods gain performance with more keywords in the evaluation. This is mainly due to the fact that the recall keeps about the same value, but the precision becomes higher, i.e. relatively more relevant relations are found and always divided by the same high number of relations.

The most outstanding method in that respect is the co-occurrences algorithm. Whereas the other NLP-methods have a precision of about 0.15, Co-occurrences can grow to 0.5. Although Hearst has a smaller number of concepts, the concepts and relations in Co-oc are better suited for the keyword-based evaluation method. When comparing the NLP-methods OpenIE and Dep++ alone, we can conclude that Dep++ performs slightly better then OpenIE. This is surprising as only a few NLP-patterns are used to generate the ontology based on the enhanced NLP-dependency relations.

Aside from the good gain in performance of Co-oc and the good performance of the keyword-based methods, we see that the combined keyword-based method is better than any of the single methods. This can mainly be explained by the higher recall: more related keywords are found when combining the methods, and therefore recall is higher. This is an indication that future work can be targeted to improvement of our ontology merging techniques on top of the union merge.

Based on these results, we consider the following possibilities for future work. First, a logical next step is improving the combination of multiple ontologies. Combining the keyword-based methods is logical as a union, but adding even more concepts and relations to the already big NLP-based ontologies might not be the best way forward. We could for example use some filtering, or seeding with keywords or an existig man-made ontology. Second, the Dep++ algorithm can be further improved by using other NLP-patterns known in the literature. This can improve the precision of this methodology, as the recall is already quite high. Third, another interesting next step is to improve on the quality evaluation method. We did some first experiments with the DOORS algorithm, but on some layers the results are not yet reliable. Using the keyword-based evaluation is objective and semantically sound, but because we use the same document set for creation and testing of the ontologies this might influence the performance.

Concluding, we made a first step towards automatically creating data-driven ontologies using a domain specific document set. We used and compared NLP-based and keyword-based techniques, and an exciting next step is to combine the best of both worlds to create even better ontologies.

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 giving useful comments and providing a representative agriculture-related document set.

References

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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.

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

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

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

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