UTRECHT UNIVERSITY

MASTER'S THESIS

A quality measure for automatic ontology evaluation and improvement

Author:
Kambiz SEKANDAR
Student number:
4142721

Daily Supervisor:
Dr. Ir. Jack VERHOOSEL
Project Supervisor:
Dr. Rosalie IEMHOFF

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Faculty of Science
Department of Information and Computing Sciences

June 26, 2018

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Department of Information and Computing Sciences

Master of Science

A quality measure for automatic ontology evaluation and improvement

by Kambiz SEKANDAR

Ontologies are a useful tool for storing information, as they allow for automated reasoning. However, creating, maintaining, or evaluating ontologies are time-consuming processes. We started with automating the evaluation process, based on a framework from semiotics. Thereafter, we show a method to improve on existing NLP algorithms for the task of generating ontologies. The main contributions of this work are a multi-criteria, automated ontology evaluator, and a proof of concept for using such an evaluator for an automatic improvement of ontologies.

Acknowledgements

I have written my thesis in fulfilment of the degree for Master of Science (Utrecht University) at TNO, Soesterberg. I want to thank my daily supervisor, Jack Verhoosel, for everything he has done to help me with my thesis. Our weekly meetings where we sparred for over an hour sometimes, all the other moments during week where I could just pass by his office and his willingness to help me with the struggles I came across, helped me produce the thesis as it is right now.

I also want to thank my project supervisor, Rosalie Iemhoff, who has volunteered to be my supervisor from Utrecht University, even though she was not familiar with my topic. She has firstly helped me by just being my supervisor as it was required for me to have one in order to start my thesis project. Furthermore, she was very willing to help me with everything she could, through thick and thin.

I also want to thank TNO, for being such a hip place to work. I enjoyed every part of being there. From the open-mindedness of all the colleagues, to the lovely garden behind the building, and from the forest on a walking distance, to all the cool places that were to explore in the building. The fun activities like the "borrels" by a committee within TNO, helped me socialize with a lot of new people. I had plenty of stuff to look forward to, and it helped me forget my project when I needed it to.

At last, I want to thank the all the other interns in our department. We had a lot of fun during the lunch breaks, at the so called "borrels" and even during work time. Sometimes, we even had some discussions related to our projects.

Contents

Abstract iii

Acknowledgements v

1 Introduction 1

2 Relevant Literature 3
2.1 Ontologies . . . 3
2.1.1 Conceptualization . . . 5
2.1.2 Ontology language . . . 6
2.2 Ontology Evaluation Methods . . . 7
2.2.1 Golden Standard . . . 8
2.2.2 Application Based . . . 9
2.2.3 Human Assessment . . . 10
2.2.4 Data Driven . . . 11
2.3 SONNET . . . 12
2.3.1 Stanford CoreNLP . . . 12
2.3.2 Hearst patterns . . . 13
2.3.3 Word co-occurrence . . . 14

3 Automated Evaluation of NLP generated ontologies 15
3.1 Quality Criteria for ontologies . . . 15
3.1.1 The semiotic framework . . . 15
3.2 Implementation of quality measures for ontologies . . . 16
3.2.1 Syntactic layer measures . . . 18
3.2.2 Semantic layer measures . . . 19
3.2.3 Pragmatic layer measures . . . 22
3.3 Results . . . 23
3.4 Conclusion . . . 24

4 Improving ontologies 27
4.1 Ontology improvement heuristics . . . 27
4.1.1 The heuristic . . . 27
4.1.2 Expectations . . . 28
4.2 Results . . . 29
4.3 Conclusion . . . 30

5 Conclusion 31

Chapter 1

Introduction

With the increasing amount of information every day, it gets harder to keep track of all available knowledge. Information is spread over the internet, however, without being structured. It is hard for both humans and computers to find relevant information. In a world where automation increases, we would like computers to easily share their knowledge. A solution for this is an ontology, which is a structured, formal knowledge graph that is interpretable by both humans and computers.

Ontologies are a useful tool for storing, adding and retrieving knowledge about a specific domain. The logical representation of information that forms the basis of an ontology lends itself to effective use by both humans and computers. However, the process of constructing an ontology is a difficult task because of the tremendous amount of knowledge we prefer to see in such an ontology. Up to a couple of years ago this task was performed solely by humans, but with all the progress in the field of Natural Language Processing (NLP) algorithms, this task can now also be supported by such algorithms.

NLP-algorithms can extract objects (entities) and relations between them from a given corpus of text, which could be used to create ontologies. The problem when constructing an ontology with such tools is that there is no guarantee that the produced ontology is a good one. A human expert would be required to look for inaccuracies in the ontology, but as these ontologies tend to become large, this could require more time than constructing an ontology from scratch. Therefore we want to construct a means of automated ontology evaluation so that we can properly evaluate an ontology as well as to improve our NLP-tools regarding the quality of the ontology they produce.

There are some inherent problems in evaluating an ontology, even when they are made by experts, since different experts of a field could create distinct ontologies considering the possibilities to model information. However there is a lot of literature on aspects that would improve an ontology. We aim for an extensive list of such features in order to make a good fit for our proposed automatic evaluation of an ontology. When we have such a list, it might not be trivial to automatically implement these, as some of the features might require a deeper understanding on the subject. For example, if you want to evaluate an ontology on a domain of interest, you need to know what is missing to be able to calculate the coverage of the ontology. Our implementation will thus be able to distinguish good ontologies from bad ones based on our set of criteria.

Since NLP-tools are not (mainly) designed for the creation of ontologies, they do not always produce an ontology that is directly applicable. In order to improve that process, an evaluation function is needed. Therefore, the following research questions will be answered in this thesis:

1. What are quality measures humans use in their assessment of ontologies, and

which of those are feasible to be implemented as quality measures for automatic ontology evaluation?

2. How can we improve existing NLP-tools by extending them with heuristic rules that implement ontology adjustments according to our quality measures?

The purpose of this thesis is to answer the questions above and thereby to improve our understanding on ontologies and NLP in general. The proposed findings will be tested to evaluate whether ontologies can be improved using heuristics based on our quality measures.

Chapter 2

Relevant Literature

In this section we discuss the relevant literature regarding the research questions as posed in the introduction. Firstly, we will dive into the world of ontologies. What are characteristics of ontologies, where lies their strength and what are some problems that occur with ontologies? These are the questions we try to answer in the first section.

We are not the first ones to try and come up with a quality measure for ontologies. We discuss the literature on this subject and will give an overview of the present ontology evaluation methods in the second section. We will see that current evaluation methods lack certain practical aspects, and therefore we conclude that there is some work to do in this field.

Furthermore, as our second question was about the improvement of NLP-tools for making better ontologies, we will discuss the tools that we used to make our ontologies. Having an insight in how they work can improve our understanding of certain characteristics in the ontology that was produced.

2.1 Ontologies

There exist several definitions for ontologies since the term is used in various fields. We will use the most common definition as used in computer science: "an ontology is a formal conceptualization of a domain of interest" [3]. Guarino [13] argues it should also be shared, as there is little use to making an ontology if the goal is not to share knowledge. This makes them a sort of representation of the world that is both human-intelligible and machine-readable [14]. This is what makes them such a powerful tool, if used correctly. It allows computers to operate based on a formal semantic knowledge representation, while allowing humans to understand the processes in the knowledge base.

Figure 2.1 represents different kinds of knowledge bases. They are divided between automated reasoning and non-automated reasoning and have been plotted against the complexity of the knowledge base. A catalog is a list of concepts, with nothing but ID's. A glossary provides natural language information about the list of concepts. A thesaurus is similar to a glossary, but with the addition of relevant terms to the specific concepts. A taxonomy also provides relations between the concepts within the knowledge base, thus making it possible to automatically reason about it. A proper ontology in addition has a set of axioms in some type of logic. We will discuss these axioms later on.

Hlomani [3] divided an ontology in four main categories: Concepts, Properties, Relationships and Restrictions. Together they form an ontology on a specific domain.

Concepts are the terms which are important to talk about in the domain of interest. If we want to have an ontology on e.g. algorithms we want to include concepts

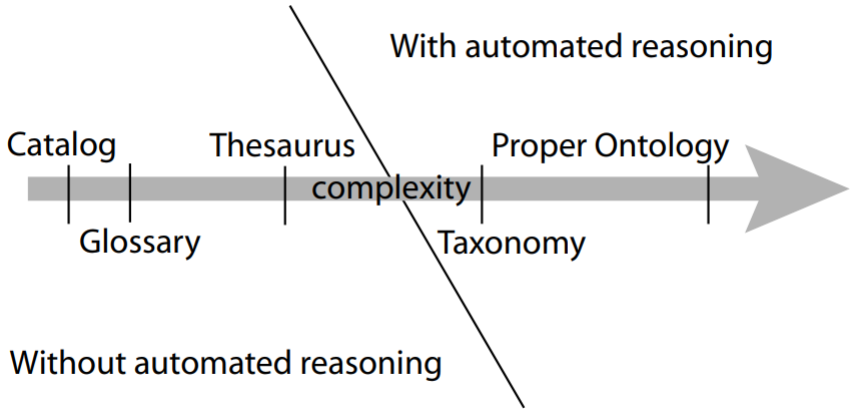

FIGURE 2.1: A semantic spectrum for ontologies. [1]

as: statistical, artificial intelligence, clustering and so on. These concepts all represent some aspect of the domain.

Properties are characteristics or attributes that concepts have. They are a characteristic that is owned by the concept, but several concepts may have the same type of properties. An example of a genetic algorithm's properties is population size among as many properties you define in the ontology.

Relationships exist between different concepts. They express the semantics between them. The most common relation is the subclass-relation, or the 'is-a' relation. An example is the 'is-a' relation between genetic algorithms and artificial intelligence-algorithms. Of course other relationships exist, such as e.g. a tire has the relationship 'a-part-of' car. Of course the concept of 'tire' can also have a relationship 'a-part-of' with a bike, but it depends on the domain of the ontology whether you would prefer to add that or leave it out. Figure 2.2 is a representation of a part of an ontology. This shows how different concepts are saved as nodes with properties, and relations between these concepts.

Restrictions (or axioms) are the smallest units of knowledge in an ontology [1]. They define formal semantic relations between concepts, on properties and on concepts. Restrictions are helpful when adding information to the ontology, in order to keep it coherent. An example of this is creating an ontology with the restriction of only having distinctive concepts, thus not allowing (seeming) synonyms. This can formally be done by adding the restriction that concepts can only be added to the ontology when they have at least one distinctive property with all the other concepts. Restrictions also allow for checking whether an ontology is coherent after it was defined.

We have seen some key components of an ontology, but its definition addresses the term conceptualization as well. So what is a conceptualization precisely? We will see that actually there cannot be a precise definition of the term since the meaning of conceptualization is vague. We will start by explaining why this is the case in the next subsection. This will give us insight in the limitations of problems of conceptualizations and therefore also of ontologies. Afterwards we will look at how some conceptualizations could be better than others. In the second subsection we will look at ontology languages, which are used to create ontologies.

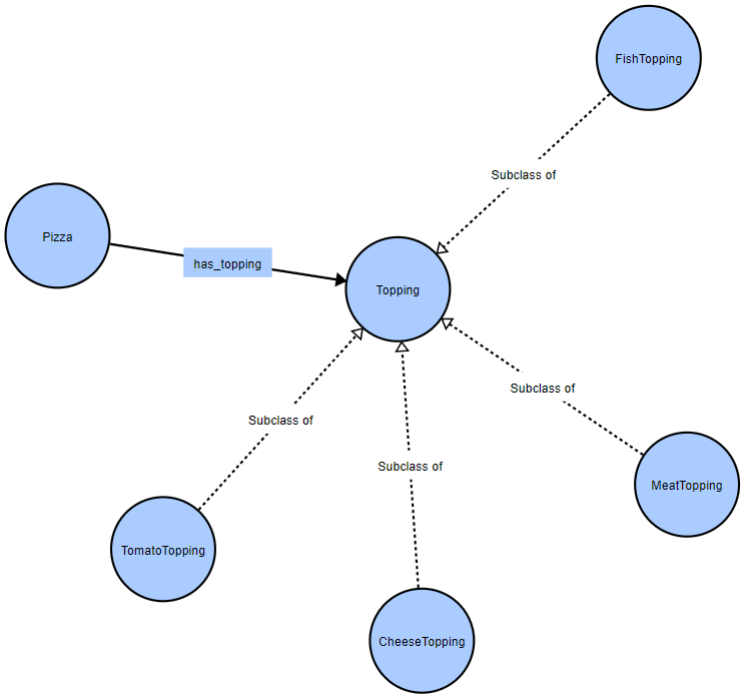

FIGURE 2.2: A part of an ontology.

2.1.1 Conceptualization

According to Guarino [13] a conceptualization is an abstract simplified view of the world that we wish to represent. Every form of knowledge base is in some way committed to conceptualizations, whether it is stated explicitly or left implicit. For example, lets assume we have an ontology on human resources from a large software company. No individual in the workplace is exactly the same, not even related employees or twins. They, for example, have different thoughts, different hair colours or different clothes. Making an ontology that captures every difference between its employees would become very large and might be too much for the use of the ontology, namely giving an overview of the work-floor. The goal of the ontology should therefore be made beforehand in order to check whether the outcome fits its purpose. A good ontology on the work-floor of a software company would probably include the roles of the employees, maybe their backgrounds, but no information about their shoe size.

However, some would argue that different ontology designers have different conceptualizations of a specific domain, and they are right. Vrandecic [1] argues everyone makes their own conceptualization. This conceptualization of a domain is internal to that person and not directly shareable. A way to externalize internal conceptualizations is by making an ontology. This ontology is then internalized by anyone who has to work with the ontology. So an ontology adds an extra level of conceptualization. However, since a conceptualization is also an abstraction or simplified view of the world, it gets easier to have conceptualizations that match with each other due to the reduced amount of characteristics to internalize. This assumes that every member of the group that works with the ontology, agrees on the given ontology. This will result in a shared ontology.

It should be noted that creating a shared ontology is not an easy task. Since every individual has its own conceptualization of a domain, no ontology will fit everyone's

conceptualizations. However, it is argued that some ontologies are better than others [13, 5]. One aspect Guarino [13] notes, is that an ontology should be dynamic. This means that an ontology that relies too much on the current state of the world, is not a good ontology. For example, on the work-floor we have different employees that work with each other, and some of them have to report to each other. So we have the concept of 'reports-to' that is a relation between various employees. See figure 2.3 for a representation of a part of an ontology for a work-floor.

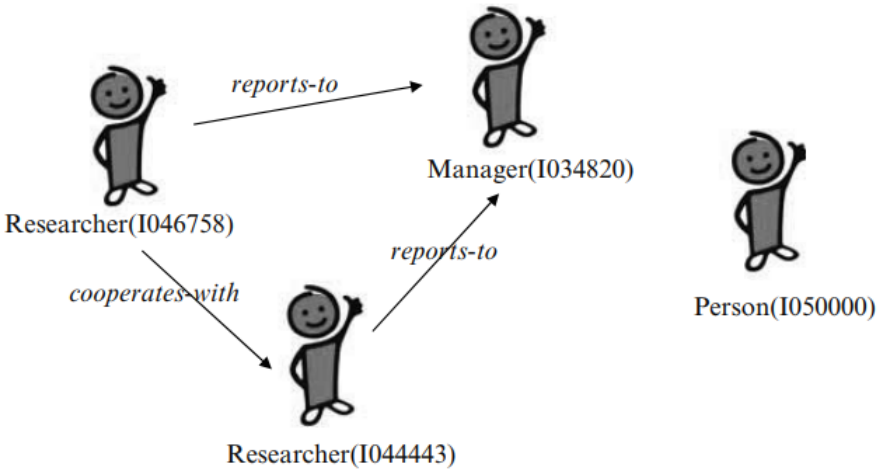

FIGURE 2.3: A small part of an ontology of a work-floor [13]

If 'reports-to' only consists of tuples consisting the individuals that have to report to another, we can see it is not dynamic when we try to add a new individual to the ontology. Every relation that involves the new individual has to be changed in order to keep the ontology up-to-date. This representation of the concept 'reports-to' does not capture its essence. The true meaning of it underlies in the fact that every 'Researcher' reports to his 'Manager'. Doing so would bring us a step closer to a dynamic, and therefore a better ontology.

We have seen the power that lies in ontologies, but also the difficulties in creating one, and inherent problems of conceptualization. A measure for the quality of an ontology is therefore wanted. However, we have also seen an intuition on the quality of ontologies. To be able to approach a measure for the quality of an ontology we will need a list of criteria. We will discuss this in the first section of chapter 3.

2.1.2 Ontology language

Ontology languages are used to construct ontologies and to be able to use them. Since ontologies are both made and used by both humans and computers, ontology languages must meet a couple of requirements [16].

- Have a compact syntax
- Be highly intuitive to humans
- Have a well defined formal syntax
- Be able to represent human knowledge

- Include reasoning properties
- Have the potential for building knowledge bases
- Have a proper link with existing web standards to ensure interoperability

Ontology languages should have a compact syntax and be highly intuitive to humans. This assures a clear overview of the ontology. Having a well defined formal syntax and the inclusion of reasoning properties allow for reasoning. Being able to represent human knowledge is of course the whole point of ontologies, as well as having the potential for building knowledge bases. Having a proper link with existing web standards are required to e.g. merge ontologies. These requirements can be difficult to achieve. In this subsection we will see what ontology languages have been used and how they dealt with those problems.

The eXtended Markup Language (XML) is widely known by the ontology community as it was one of the first languages to separate the markup of web content from web presentation. It consists of attribute-value relations, unfortunately lacking semantics. Therefore, it is not useful tool any longer for ontology creation, as it does not meet all the requirements for ontology languages.

The Resource Description Framework (RDF) was developed by World Wide Web Consortium (W3C) as the standard for web metadata. RDF uses *object-attribute-value* triples, also known as statements. Since all object are independent entities in this structure, it provides natural semantic units. This allows for reasoning systems to be build for RDF. RDF schema (RDFS) are a layer on top of the RDF model. RDFS provides developers with a tool to define a particular vocabulary for RDF data. This contributes to a type system for the RDF model.

Ontology Web Language (OWL) has more expressive power than XML and RDF(S). It is enriched with logical connectives and quantifiers, such as conjunction, disjunction, existentially and universally quantified variables. Unfortunately, this expressiveness has its drawbacks. It leads to less compact syntax, less intuitive for humans and reasoning is less time-efficient. However, these trade-offs are preferred over the use of XML and RDF(S) by ontology experts.

Reasoning within these ontology languages was one of the requirements. SPARQL is a query language, which allows the user to ask queries to the ontology [18]. It works for both RDF(S) and OWL. Some examples of information the user can retrieve are: objects with specific relations, information about a specific object or a boolean to a query.

2.2 Ontology Evaluation Methods

Ontologies are being made over a wide variety of domains by both humans and computers. They produce different ontologies and humans can mostly tell which of these are regarded as good, and which are not. However, an assessment by humans may not always be possible. It could be too expensive time-wise, or there are no available experts on the domain. Since ontologies are complex in their structure, they are better evaluated on the different criteria then trying to evaluate the ontology as a whole [9].

The evaluation of ontologies can be divided in four different approaches:

- Golden standard
- Application based

- Human assessment
- Data driven

Each of them has its own pros and cons, with respect to time-consumption, effectiveness and several other attributes. We discuss these four different approaches in the next couple of subsections, explaining how they work and why we should or should not use them.

Each method of ontology evaluation has its deficiencies. In table 2.1 we see an overview of different methodologies to evaluate ontologies [9]. A cross is put if the specified approach is capable of evaluating the specific level.

	Approach to evaluation			
Level	Golden standard	Application-based	Data-driven	Assessment by humans
Lexical, vocabulary, concept, data	x	x	x	x
Hierarchy, taxonomy	x	x	x	x
Other semantic relations	x	x	x	x
Context, application		x		x
Syntactic	x			x
Structure, architecture, design				x

TABLE 2.1: An overview of approaches to ontology evaluation [9]

Lexical vocabulary, concept, data is about whether the concepts of the specified domain are captured. Hierarchy, taxonomy takes the 'is-a' and other hierarchical relations in account. Other semantic relations is for all the non-hierarchical relations and can be evaluated separately. Context, application is a more practical level as it is evaluated within a web of ontologies, or in a specific application. The syntactic level checks for syntactic coherence within the ontology, like e.g. whether there are no loops present in definitions. Structure, architecture and design are for checking whether the ontology meets specific requirements for the structure within the ontology.

2.2.1 Golden Standard

The golden standard evaluation method works by checking the ontology that has to be evaluated with the golden standard. This approach is mostly used for ontologies that are not handmade, to be able to evaluate the ontology learner. The ontology is evaluated by comparing it to the golden standard, of which we know that it is an optimal ontology on the domain. The more the ontology differs from this golden standard, the lower its score is. Golden standard evaluation is a form of automated ontology evaluation, as the relevant information can be extracted automatically. The comparison between a golden standard and another ontology can be made for multiple criteria.

Dellschaft [12] proposed three requirements for golden standard evaluation. The first criterion is that the evaluation is based on multiple dimensions. This allows the user to perform a weighted evaluation. The weightings may be suited to the

beliefs and preferences of the user. Basing an evaluation on multiple dimensions helps in analysing the strengths and weaknesses of an ontology. However, there is an underlying criterion for the first criterion of Dellschaft. A dependency between the multiple dimensions should be avoided, in order to secure the outcomes of the evaluation are in relation with the used dimensions, since a dependency between the dimensions could result in a biased outcome of the evaluation.

The second criterion is that the effect of an error should be proportional to the degree of the error. For example, when looking at the lexical vocabulary of an ontology, big differences between multiple concepts should weigh more than a small mistake/typo.

The third criterion is for dimensions on a closed scale interval (e.g. [0...1]). a gradual increase of the error should lead to a similar gradual decrease of the score. Fulfilling this criterion helps distinguishing a difference between slight and severe errors.

Golden standard evaluation is a powerful method, as it allows for multidimensional evaluation of ontologies in an automated way. This makes it a time-efficient method. The main problem when using golden standard evaluation is the golden standard ontology. Firstly, you have to acquire one. Where will you construct it? One could get experts to construct the golden standard. But then, how are you going to evaluate it? There is no satisfactory way of solving this problem, as it is suggested that several ontologies could be equally good [9].

2.2.2 Application Based

Application based ontology evaluation tests an ontology within an application, as the name suggests. It assumes that the ontology is directly applicable within an application, and that application is in its turn to be evaluated. The evaluation of this application would be one that is easier than the evaluation of an ontology, since it would not make sense otherwise to evaluate a more complex system in a way that is not easier. Application based ontology evaluation can be done both automatically and manually depending on the type of the application and the types of the outputs of the application.

Porzel describes an application based in four components [19]. See figure 2.4 for a high level hierarchy. The application that is tested is part of a task, with a result and answers to that result. One or more ontologies are counselled by the application to result in output. Gold standard should not be confused with golden standard ontology evaluation as it is the correct answer model to the output of the task. The results of the application are checked with the gold standard and lead to performance results. Since the task is fixed, the performance results can be assigned to the ontologies.

This is a promising approach, since the complexity of an ontology evaluation method can be reduced to the complexity of evaluating a specified task. However, this method also has its downsides. This method requires a specific task to be at hand, whilst this is not always the case. Some type of ontologies are not so easily applicable within a task as others, as every ontology has a different use. An example of this is the evaluation of an ontology on a speech-recognition problem, which is easily plugged in an application, whereas many other ontologies do not have such an application to plug into. Ontologies should be easily shared and understood by both humans and computers. However, this type of ontology evaluation might fuse the ontology within a black box of the task.

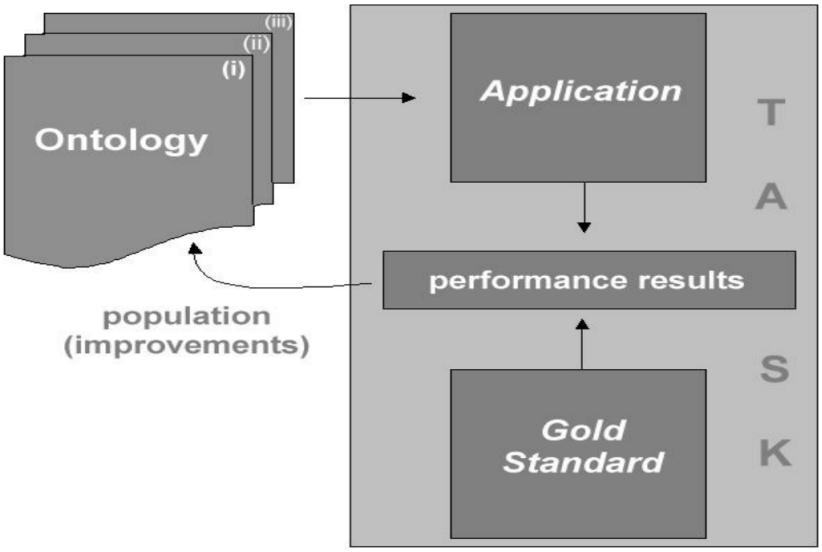

FIGURE 2.4: An application based ontology evaluation hierarchy [19]

Another main downside of application based ontology evaluation is that this type of evaluation only tests one possible aspect of an ontology, even if it has multiple uses, like most ontologies do. This will results in a one-sided evaluation of an ontology which could lead to an incorrect overall measurement. Therefore, the application based method is useful when you want to evaluate an ontology that is made with the purpose of only ever going to be used within the given application. However, the statement of an ontology being made for one purpose, is already against the nature of ontologies and therefore this method is too limited to be functional.

2.2.3 Human Assessment

Human assessment of ontologies can be divided in two main categories: qualitative and quantitative human assessment. The qualitative version consists usually of one domain expert evaluating the ontology using subjective remarks on the ontology as well. Quantitative human assessment of ontologies consists of a group of people with general knowledge of the domain evaluating the ontology. The average of this group is taken as the evaluation of the ontology. Both methods require some list of criteria for the humans to assess the ontology. An example is the criteria used in figure 2.1. In that figure we saw that human assessment is the only evaluation method capable of assessing every single criterion from that list. One reason for this may be that humans represent language in a way that is capable of reasoning on that information.

Qualitative evaluation usually requires one expert on the domain to evaluate the ontology. The assessment happens on a predefined list of criteria. The expert of the domain is capable of noticing inconsistencies and incorrect or missing information due to his knowledge. Based on these deficiencies in the ontology, a score is assigned to each criterion and a weighted sum of all these criteria is used to assign an overall score to the ontology. In theory, this method works very well, as the expert has all the knowledge to assign a score to the given ontology, and even fixing mistakes that were made by the designer of the ontology. However in practice, domain experts are costly thus making it an expensive job to evaluate an ontology in this way. For larger ontologies it might be even more difficult since the expert would have to look

at all concepts and its relations within the ontology for evaluation. Therefore, this method remains as a practically impossible evaluation method for large ontologies.

Quantitative evaluation is done by letting multiple people, with general knowledge of the domain, assign a score to the ontology in the same way as with qualitative evaluation. However, this method does not require a domain expert which makes this a somewhat more feasible option. It is of course still a costly way of evaluation since you need a group of people to evaluate an ontology. It does however solve a part of the problem of a big ontology since multiple people can evaluate smaller parts of the ontology, adding up to the whole. The score is calculated as the average of every single evaluation. Despite having some advantages over qualitative evaluation, this method still has its downsides. People might have to evaluate on parts of a domain that they have no knowledge about. This would level out if the majority of the group does possess that knowledge, but that is not always the case. This method still is not a practical solution for ontology evaluation as the cost of the evaluation process cannot be reduced.

Human assessment as a method for ontology evaluation works well in theory, since humans represent language in a way that allows for reasoning, and therefore for evaluating ontologies. This method of evaluation works well for smaller ontologies where the costs of humans tend to be smaller. However, for larger ontologies, human assessment is too costly to compete with other ontology evaluation methods.

2.2.4 Data Driven

Data driven ontology evaluation is an automatic evaluation method that is similar to human assessment. This method will score the given ontology on a predefined set of criteria. However, humans needs to have some degree of expertise to assess an ontology, so how is an automated ontology evaluation method going to address that problem? The approach to this problem is adding data to the evaluation process. This data consists of natural language text documents on the domain. The main issue with an automated evaluation method for ontologies is the lack of context. An automated evaluator needs to be able to place the ontology in a context in order to be able to spot e.g. essential missing information. The context that a simple algorithm lacks when trying to evaluate an ontology, is compromised by extracting the relevant information from the data.

Brewster [5] came up with a first working version of a data driven ontology evaluation methodology. It scores an ontology on one single criterion, namely the overlap between the entities in the ontology and the extracted terms from the data. These terms are extracted by latent semantic analysis [20] to identify keywords from the data. These keywords are expanded by using WordNet to add two levels of hypernyms. In the final step the entities from the ontology are mapped to the expanded keywords from the data in order to score the ontology.

Data driven ontology evaluation is very powerful since it can assess ontologies in the same way humans do, by extracting domain knowledge from data. However, this particular data driven approach is of course limited, since this method only takes one criterion into account.

Vrandecic [1] and Hlomani [3] both compiled a set of criteria from literature, in order to evaluate ontologies automatically based on these criteria. However, Their approach has not been implemented (publicly) as some of the criteria defined were too complex to be automatically calculated. Their research was therefore mostly of use for human assessment.

A recent approach by McDaniel et al [21] describes a data driven ontology evaluation method based on semiotic theory. This approach is fully automatic and an implementation is available on the web. This evaluation method uses 14 criteria divided in four layers: syntactic, semantic, pragmatic and social. The overall quality is the sum of the weighted values of each layer:

$$Quality = w1 * S + w2 * E + w3 * P + w4 * O \tag{2.1}$$

Where in equation 2.1 S is for the syntactic layer, E for semantic, P for pragmatic and O for Social. The weights $w1$ to $w4$ can be adjusted to the preferences of the user, but have been distributed proportionally by McDaniel et al. This evaluation method does however require as input a set of ontologies instead of a single ontology, since some of the criteria are evaluated by extracting information from multiple ontologies. The ontologies are ranked relatively and given a score, but since the method depends on information from all ontologies, it will assign a different score to the same ontology if the original set was different.

Data driven ontology evaluation is an automated method for evaluating ontologies. It can compensate for the lack of domain knowledge by extracting information from data and evaluate ontologies in a similar way as human domain experts would do. This evaluation method is however more useful as data driven ontology evaluation would only take seconds for smaller ontologies and can be performed on every possible domain, whereas human assessment requires time at least linearly to the complexity of the ontology and domain experts are usually scarce.

2.3 SONNET

Approaches to Natural Language Processing (NLP) are able to extract information from natural language. This is useful for constructing ontologies, since NLP tools can retrieve concepts, their properties and relations between concepts within the domain of the natural language source. At TNO there is a first version of a Semantic Ontology Engineering Toolset (SONNET). At this moment SONNET contains Stanford CoreNLP [22], HearstPatterns [23] and word co-occurrence [24] as its NLP tools. In this section we will briefly discuss these tools to get more insight into the generation of ontologies trough NLP tools.

2.3.1 Stanford CoreNLP

Stanford CoreNLP is one of the most used tools for analysing natural language. The authors suggest that this follows from its simple design, little requirements from the user and the robustness and quality of the analysis. Furthermore it has an online tool and it is open source [22]. Next to support for the English language, it has (partial) support for Chinese, Arabic, French and German.

Stanford CoreNLP can be seen as a pipeline through which the data flows. First the text is splitted up in sentences and words (tokens), and cleaned from irrelevant text (such as XML-tags). Tokens are then labelled with their part-of-speech(POS) tag and named entities are recognizede by a named entity recognizer (NER). Some named entities are Person, Location, Date, etcetera. After preprocessing, the full syntactic analysis is provided, which connects the tokens of the sentence. The entire co-reference graph of a sentence can be seen in figure 2.5. This figure shows

the dependencies between every word, and its relations within the sentence. Adjectives are correctly connected with their corresponding noun, and the verb 'jumps' correctly takes a subject on its left-side and an object on its right-side.

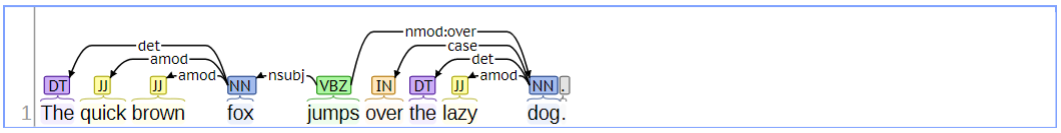

FIGURE 2.5: Snapshot from the online Stanford CoreNLP tool

OpenIE is the information extraction annotator of Stanford CoreNLP toolset and is used as the next step in the generation of the ontology. It extracts all entities and its relations from the dependency tree as shown in the previous figure. A representation of all the entities and its relations can be seen in figure 2.6. They are saved in JSON format ontology, which consists of a relation, a source and a target. In this case the source would be 'fox', 'brown fox' and 'quick brown fox'. The target would be 'dog' and 'lazy dog', and the relation between those entities is 'jumps over'.

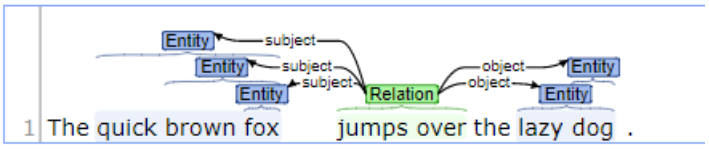

FIGURE 2.6: Snapshot from the online Stanford CoreNLP tool

This however might seem unintuitive to some, as they could argue that 'fox' should not appear in three different entities, as they all represent the same fox. The adjectives should, for example, be added as properties of the entity. This is a known problem, but it also is a practical example for the use of ontology evaluation.

2.3.2 Hearst patterns

Hearst patterns are a way of extracting taxonomical relations from an unrestricted text source [4]. It is able to retrieve hyponym relations from text, such as "England-Country" where "England" is the hyponym of "Country". Such relations are useful for ontology creation as a hyponym relation represents an "is-a" relation between terms.

Natural language often has recurring lexical structures as: "*NP*, such as *NP*", or "*NP*, or other *NP*". Hearst patters looks at these patterns and extracts the hyponym relations between the noun phrases. Some of these patterns follow from common-sense, but there is also a procedure to find new patterns.

1. Decide on a lexical relation that is of interest, e.g. the "is-a" relation
2. Gather a list of words for which it is known that it holds, e.g. "England-Country"
3. Find instances of these words that are syntactically close
4. Find the patterns across these instances and hypothesize them
5. If a pattern was confirmed, go to step 2 and gather more instances of the target relation

This procedure is not yet automatically feasible, but since usually the same structures are used in natural language, the list of found patterns remains manageable. Hearst patterns currently consist of a list to extract certain taxonomic relations, but this list will not find every instance of the target relation in the text as some of the instances are hidden in more complex structures. However, Hearst patterns are capable of finding the majority of the taxonomical relations, thus justifying its use for ontology creation despite its low recall. Therefore, an ontology based solely on Hearst patterns might not result in a good ontology. However, it is a good method to extract explicit hyponym relations from text. Hearst patterns are able to provide an addition to other ontologies, for example those made by other NLP-tools.

2.3.3 Word co-occurrence

Word co-occurrence is a method to find words that might have a relation with one another. Words that co-occur probably have some relation [24]. Firstly, a matrix is built of all the words, except for e.g. a list of stop words which the user would not want to include. For every combination of two words it is counted how many times these words co-occur. Co-occurring can be defined arbitrarily, e.g. at most 2 words apart from each other. Co-occurrence between the same two words is not defined. Figure 2.7 is an example word co-occurrence matrix with a maximum distance of 1 (so they have to be next to each other). The used corpus consists of the sentences: *"I enjoy flying. I like NLP. I like deep learning."*

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \end{bmatrix} \end{array}$$

FIGURE 2.7: An example word co-occurrence matrix [25]

Words that are used together often, probably have some sort of relation between them. When creating ontologies, one is interested in concepts and relations between those concepts. Word co-occurrence can solve part of the problem by finding concepts that have an undefined relation.

Word co-occurrence is used to create ontologies by setting relations between highly co-occurring terms. However, this relation remains undefined by the algorithm, thus making this approach to ontology creation on its own doubtful. A combination with another algorithm that finds relations between two words could be a possible solution to this problem. On the other side, this approach does make a good selection of relevant terms to a text source, which is the main reason for implementing it in the Sonnet platform.

Chapter 3

Automated Evaluation of NLP generated ontologies

Ontologies can be produced relatively quick by NLP-tools compared to ontology creation by humans. However, there is not a satisfying approach to measuring the quality of these NLP generated ontologies. This evaluation process must be accomplished in an automated way as well, since human evaluation would take too much time. In this chapter we will describe our approach to automated ontology evaluation.

We will start by introducing the semiotic framework in which we will implement our quality criteria of ontologies. This framework is based on the study of signs, and it is layered in a couple of different groups which each in turn touches upon an aspect of the ontology.

In the second section we will look into the mathematical implementation of our approach to ontology evaluation. We also implemented this in Python and tested it on different types of ontologies created in the SONNET platform. The results of this implementation will be shown in the third section of this chapter.

3.1 Quality Criteria for ontologies

3.1.1 The semiotic framework

In chapter 2 we discussed different types of ontology evaluation. We saw that humans use multiple criteria in assessing the quality of an ontology. Our data driven approach resembles human assessment in its use of multiple criteria. Our approach is based on the theory of semiotics, which is the study on meaning-making (understanding) and sign process. Semiotic theory says that the meaning of written text is divided in multiple layers. See figure 3.1 for the different layers of this framework.

This structure is of a hierarchical form. The upper layers depend on the layers below them. The bottom two layers are two most important layers, but also the most meaningless ones. Without physical signs of any form, meaning could never be attached to words as they would not exist without the physical world and without a meaning (semantic layer), it can never fulfil its intended purpose (pragmatic layer). We assume the physical and the empirical layer to be as they are in our ontology evaluation model, since we work with a fixed set of symbols and use the same ontology format everywhere. We use criteria that are representative of different layers to evaluate ontologies on different aspects.

The social layer for ontologies depends solely on its use in a community, so this layer could only be evaluated after an ontology has been used for a while [21]. In this thesis we also want to evaluate NLP-generated ontologies before they will be used, in order for them to have the ability to be revised and improved. Therefore,

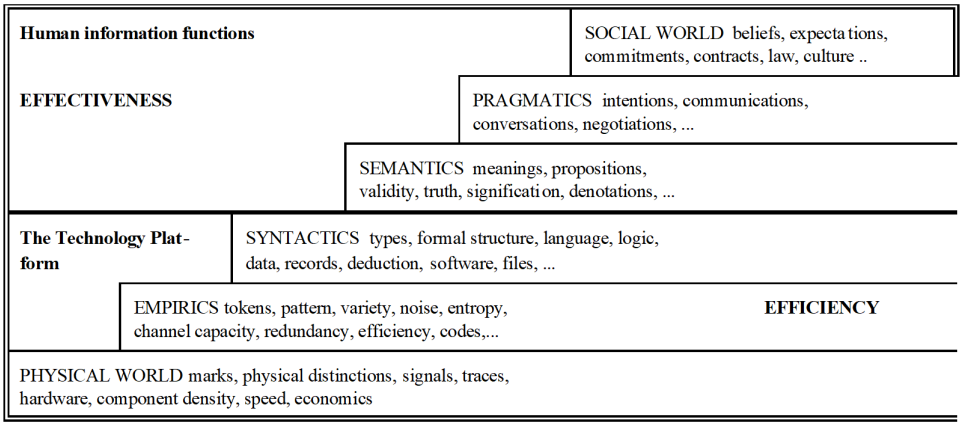

FIGURE 3.1: The different layers in the semiotic framework [27]

Layer	Metrics	Source
Syntactic	Structure	[21]
	Richness	[21]
Semantic	Precision	[5]
	Variance	[26]
Pragmatic	Clarity	[28]
	Adaptablity	[21]

TABLE 3.1: Quality measures

we will leave out this layer as well. McDaniel et al 2.1 have included this layer in their quality framework for ontologies. However, they implemented it by, e.g. using counts for the amount of times it was cited. This has two problems for NLP-generated ontologies. First of all, this requires meta-tags on the ontologies, as e.g. the number of citations is not at hand directly from an ontology. Secondly, NLP-generated ontologies have not been implemented in a social environment yet.

The syntactic layer within this framework is about the formal structure of an ontology. Semantics is about the meaning of the concepts within in an ontology and their relations. The pragmatic layer raises the question of how usable an ontology is.

3.2 Implementation of quality measures for ontologies

We have developed a quality measure for ontologies based on the semiotic framework described in the previous section. Table 3.1 shows the metrics we have used for calculating the quality of an ontology, and in which semiotic framework layer they fall. The metrics were derived from prior work. All of these metrics have been evaluated in order to only add components that have a grounded foundation, mathematically speaking. Doing so, ensures us of the quality of the quality function, but it may be lacking on some aspects. The overall quality of an ontology is computed as the weighted sum of each layer, and each layer is computed as the weighted sum of each of its metrics.

$$Quality = (Syntactic + Semantic + Pragmatic)/3 \tag{3.1}$$

The **syntactic layer** is concerned with the structure of the relations between concepts. Without a coherent structure, an ontology would not be able to be parsed, making it impossible to retrieve knowledge. An ontology is a tree like structure, where the root-node is the domain of the ontology, and the other nodes are pieces of information within that domain. This tree structure should be evenly distributed, as e.g. a tree with one very deep branch and only short ones for the remaining branches suggests that this ontology misses some key information. However, certain branches could of course be of less importance than other branches, so this has to be taken in account when defining quality measures for the syntactic layer.

The goal of evaluating the syntactic layer is to find shortcomings of the ontology, in order to be able to find suggestions for improving the ontology. The original work of McDaniel et al included a measure for lawfulness in its ontology evaluation[21]. Lawfulness is the criterion that checks whether the ontology is written in the correct syntax, as it can not be read otherwise. If an ontology was parsed successfully, it would receive a score of 1 for that measure. However, we leave out that measure as ontologies that can not be parsed, can not be evaluated. Thus, such a measure does not provide any useful information for improving the ontology.

Structure and richness are the measures we used for evaluating the syntactic layer. The structure measure looks into the class and subclass relations of the ontology. A higher score for structure can be attained when the information is distributed well, in a syntactical sense. For example, we try to punish ontologies with one very deep branch and limited branches for the other concepts with this measure. Richness evaluates the ontology on the syntactic information within classes and relations. Thus, an ontology with lots of different relations makes use of a broader variety of relations between concepts. Using these two criteria for the syntactic layer, we cover up most syntactical aspects of an ontology.

The **semantic layer** is about the meaning of the ontology. Can users, humans and machines, understand this correctly? Furthermore, do the concepts within the ontology map to the real world? Is the ontology complete, regarding the knowledge it represents? If one of these is answered no, then the semantic layer of an ontology should score lower in the evaluation process.

The task of representing 'correct' semantics is a hard one. As we saw in chapter 2 in the section about conceptualizations, there is no single correct representation of the world. That makes it hard to measure the semantics of an ontology in precision and recall. However, we also saw that there are certain representations that are better than others. So in this layer of the semiotic framework we will try to distinguish ontologies that are lacking in their representation from ontologies that do not.

One of the measures we use for the semantic layer is the data-driven ontology evaluation approach by Brewster [5]. This method maps the concepts in an ontology to keywords from a corpus on the domain of the ontology. The amount of successful mappings between the ontology and a corpus is the fit regarded as the semantic quality of an ontology. This method is an estimate of the precision of the concepts within the ontology.

In an attempt to measure the accuracy of an ontology, Sanchez et al [26] invented an intuitive measure for ontology accuracy. This measure, called semantic variance, relies on work by Fernandez et al [29], which suggests that a taxonomic heterogeneous structure is correlated with a high semantic accuracy. The taxonomic structure is the arrangement of the concepts within an ontology. Intuitively, this seems to be logical, as the taxonomic structure of an ontology evolves itself during the process of generating the ontology and such a process rarely produces an evenly distributed knowledge structure. The semantic variance is measured by calculating a taxonomic

distance to the rootnode for each node in the ontology. We will dive into the exact calculation of this measure in subsection 3.2.2 in more detail.

The **pragmatic layer** is about the usefulness of an ontology. In order to be useful, the knowledge represented in the ontology must be presented as clear and complete as possible. Ontologies end up being used for other goals than the intended purpose[2]. Since ontologies always undergo changes it is useful to have an ontology that has an ability to adapt for new purposes[1]. Changes are always necessary as information about the domain can change, or the application in which the ontology is used, may change.

Ontologies must exhibit knowledge in the most clear sense. An ontology with a lot of vague terms can cause disagreement among the users of the ontology. According to Alexopoulos and Mylonas, vague terms are domain specific and should be classified as vague by domain experts [28]. In our automated quality measure for ontologies, we try to estimate the vagueness of concepts by checking each node for its use of adjectives and adverbs, as they are mostly instances of a broader concept, and therefore should be properties of the concept.

The adaptability criterion checks whether the ontology is able to respond to changes. Can new information easily be inserted in the ontology? For this criterion we look at the ratio of leaf-nodes to the whole ontology and the depths of the leaf-nodes. For example, if the average leaf-node depth is close to the maximum leaf-node depth, it assures a secure foundation to built upon.

3.2.1 Syntactic layer measures

We define two different criteria in our approach for the syntactic layer in the quality function based on the semiotic framework. These criteria are richness and structure. The overall quality for the syntactic layer is calculated by the weighted sum of the two.

$$Syntactic = (Richness + Structure)/2 \tag{3.2}$$

The first criterion, richness, is an estimate for the amount of information that is in the ontology, on top of the basic relations between concepts. Richness refers to the structural proportion of features in the ontology. For example, the more properties a concept has, the higher the richness measure should be. Properties are the set of relations between nodes, so that every different relation only occurs once in the set. This measure is divided in two types of richness: attribute richness, and relationship richness.

$$Richness = (Attribute\_Richness + Relationship\_Richness)/2 \tag{3.3}$$

Attribute richness aims at calculating the amount of properties per class. McDaniel et al calculate this measure by dividing the amount of properties by the amount of nodes in the ontology [21]. However, this could lead to scores higher than 1 for the attribute richness criterion. Since we want to scale this measure to a maximum of 1, we use a function that is limited between 0 and 1 with a higher ratio of properties per class leading to a higher value of the attribute richness. In equation 3.4, *Att_Ratio* is the ratio of properties per class. This is calculated by simply dividing the amount of properties by the amount of classes.

$$Attribute\_Richness = (2^{Att\_Ratio} - 1)/2^{Att\_Ratio} \tag{3.4}$$

The second type of richness is the relationship richness. McDaniel et al argue that an ontology should make use of the richness of the English language by defining more than just the relations between concepts in an ontology [21]. Here we take the ratio of properties and divide it by the total amount of relations in the ontology. The difference between properties and relations is that properties are a set and relations are a list of properties. For example, we can have an ontology where only one and the same property is used ten times as a relation between different concepts.

$$Relationship\_Richness = N\_Properties/(N\_Properties + N\_Relations) \tag{3.5}$$

The last criterion in the syntactic layer is structure. We aim at a tree like structure for ontologies, so what we want is a structure with a high ratio of subclasses compared to the amount of root nodes. A root node is defined as a concept which is not the target of any relationship in the ontology. In graph theory this is the equivalent of having no incoming arrows. The calculation for structure can be found in equation 3.6.

$$Structure = (N\_Nodes - N\_Rootnodes)/N\_Nodes \tag{3.6}$$

Our syntactic layer of the evaluation framework thus consists of both syntactical relations on the level of the nodes and the level of relations. This assures us that, as long as the ways of measuring the criteria within this layer are correct, that this layer is complete.

3.2.2 Semantic layer measures

The semantic layer consists of the criteria precision and accuracy. Usually, accuracy is a measure that is harder to find than recall, since you also need to know the amount of true negatives to calculate it. However, our method for calculating accuracy is an estimation, thus we do not need to know those exact numbers. We calculate the semantic layer as the average of the two criteria, as in equation 3.7.

$$Semantic = (Precision + Accuracy)/2 \tag{3.7}$$

The first method is called precision. If we try to analyse this method in terms of true positives, false negatives, etc. we see that a word in the ontology that maps to a keyword from the corpus is classified as a true positive. A concept that was not mapped to a keyword from the corpus is counted as a false positive. Since we divide the amount of true positives by the total amount of concepts in the ontology, we get an estimation of the precision of the semantic layer of an ontology.

We have used the data driven approach by Brewster et al to calculate the precision [5]. This approach is performed in three steps.

1. Identifying keywords
2. Keyword expansion
3. Ontology mapping

In the first step, we extract keywords from a corpus using the TextRank algorithm [30]. This algorithm is a graph based ranking model in the domain of text processing. It ranks words within a corpus. We retrieve the keywords by taking the top 1% from this list. This percentage was a good fit for the size of our ontologies

and the corpora. However, we could let it depend from the size of the ontology to assure bigger ontologies can also score well on this criterion. In the second step, we expand our extracted keywords. Since terms in an ontology are compact representations of the concepts, they can be represented in multiple different ways. We used WordNet[32] to add two levels of hypernyms to each term in the ontology. In the last step we map the terms from the ontology to the extracted keywords from the corpus. The higher amount of terms that fit the corpus, the higher the precision will be. The exact calculation is done by dividing the amount of mapped terms by the total amount of terms within the ontology, as can be seen in equation 3.8.

$$Precision = N\_Mapped / N\_Nodes \tag{3.8}$$

As suggested by Fernandez et al, structural features can be used to estimate the accuracy of an ontology. Good predictors are depth and breadth [29].

- Maximum depth (length of the longest taxonomic branch in the ontology, measured as the number of concepts from the root node to the leaves of the taxonomy), average depth (average length of all taxonomic branches) and depth variance (disper- sion with respect to the average depth, computed as the standard mathematical variance).
- Maximum breadth (width of the taxonomic level of the ontol- ogy with the largest number of concepts) and breadth variance (dispersion with respect to the average breadth).

Sanchez et al underlines these measures as good predictors for accuracy. Others, as e.g. the amount of nodes, did not show a significant relationship with the accuracy of an ontology [26]. This implies that adding everything we know to an ontology, does not increase the accuracy of it as the amount of mistakes increases by doing so. The underlying problem is of the same form as with conceptualizations, where we can have multiple representation, which will never be exactly the same.

The approach by Sanchez et al to estimate the accuracy is based on the semantic variance of the taxonomic model of an ontology [26]. The definition for semantic variance is the squared semantic distance between each concept c_i and the root node of the ontology. If there are multiple root nodes (multiple trees), then an additional root node is created, with arrows to all previous root nodes.

$$Semantic\_Variance = \left(\sum_{c_i \in C} d(c_i, Root\_Node)^2\right) / |C| \tag{3.9}$$

In equation 3.9 d() is the function that calculates the semantic distance between two nodes, and |C| is the cardinality of the set of concepts, excluding the root node. The calculation for semantic variance is very similar to calculating variance elsewhere. The main part of this formula is the formula for the semantic distance. So we start by looking at that formula first before evaluating this criterion.

When defining the taxonomic distance we should consider the breadth and depth of an ontology, as they are good predictors for accuracy. A distance function based on edge counting formulas is therefore not what we want, as such a formula is dependent on the size of the ontology, and we have seen that the size of an ontology is not an indicator for accuracy. The semantic distance between two nodes is calculated by Sanchez et al as the differences in the parent sets of these nodes [26]. Exactly the union of the parent sets is subtracted by the intersection of the two parent sets, and divided again by the union. We take the log to scale between 0 and 1, non-linearly.

See equation 3.10 for the exact formula. We define a parent set of a node (*T(node)*) as all of its precedents in the tree, including the node itself.

$$d(c1, c2) = log_2(1 + \frac{|T(c_1) \cup T(c_2)| - |T(c_1) \cap T(c_2)|}{|T(c_1) \cup T(c_2)|}) \tag{3.10}$$

We have seen the formulas for calculating the semantic variance, but it might be hard to grasp the simple idea that lies beneath. We will show this by using a small example. In figure 3.2 we see a small sample ontology, with one root node and four sub-concepts. For filling in the formula for semantic variance, we sum over all the concepts, except for the root node and then calculate the distance between the root node and the given node. So in this case, for each of the nodes the difference in the ancestor-sets of the root node and the given node is equal to $\frac{1}{2}$. The cardinality of the set of concepts, excluding the root node is 4. The filled-in formula is in equation 3.11.

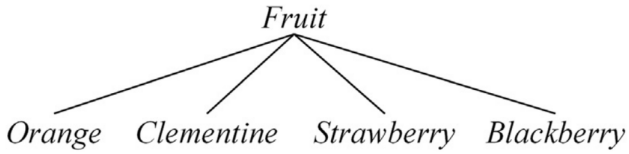

FIGURE 3.2: A sample ontology O_1 [26]

$$Semantic\_Variance(O_1) = \frac{4 * (log_2(1 + \frac{1}{2}))^2}{4} = 0.342 \tag{3.11}$$

We see in the sample ontology O_1 that the concepts below the root node are indistinguishable from each other. They all share exactly the same set of properties, except for their name id's. Therefore, we perceive a low score for the semantic variance of this ontology, which is intuitive.

The sample ontology O_2 has two extra sub-layers for the four leaf nodes. These two extra nodes will also have to be taken in account for when calculating the semantic variance. The function is as in equation 3.12.

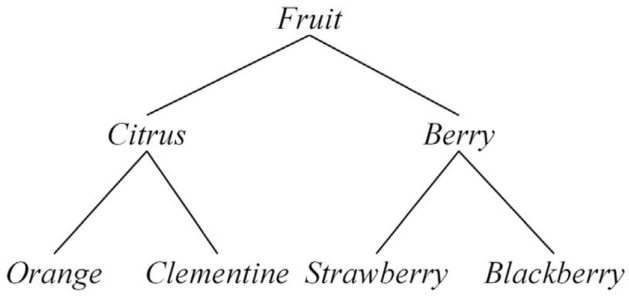

FIGURE 3.3: A sample ontology O_2 [26]

$$Semantic\_Variance(O_2) = \frac{4 * (log_2(1 + \frac{2}{3}))^2 + 2 * (log_2(1 + \frac{1}{2}))^2}{6} = 0.476 \tag{3.12}$$

Since there is some degree of higher variance in sample ontology O_2, we see that it receives a higher score for semantic variance. The more complex the ontology gets, the higher the semantic variance will be. With more complex we mean less

homogeneous in the taxonomic structure. This measure has proven to be a good indicator for the accuracy of an ontology.

The semantic layer of an ontology is about the meaning of the ontology, and its concepts and relations. In this layer we measure the mapping of the concepts to keywords from a corpus on the same domain of interest. Furthermore we use a criterion that estimates the accuracy of an ontology, by measuring the amount of information that can be retrieved from the taxonomical structure from the ontology.

3.2.3 Pragmatic layer measures

For calculating the pragmatic aspects of an ontology we have defined a clarity and an adaptability measure. The clarity measure says something about the terms used in the ontology. Adaptability looks more on the side of properties for reusing an ontology. See equation 3.13 for the formula for calculating the pragmatic layer.

$$Pragmatic = (Clarity + Adaptability)/2 \tag{3.13}$$

The idea for calculating clarity comes from Alexopoulos and Mylonas [28]. They define a vagueness-oriented quality measure for ontologies. They calculate vagueness on different levels, but the main work still has to be done by humans, namely assessing the terms that are vague. This requires human experts in the domain of interest. Our approach for calculating clarity is completely automated, and therefore requires no human experts.

Instead of using human experts to determine vague words, we use NLP-tools to look for them. We define a vague term, as a term which consists of 1 or more words being an adjective or an adverb. We use WordNet to assess which terms consist of at least one adjective adverb. Usually, terms consisting of such words are vague. For calculating the clarity, we divide the amount of vague concepts and relations, by the total amount of concepts and relations. We subtract this ratio from 1 to get our clarity value, as shown in equation 3.14.

$$Clarity = 1 - (N\_Vague/(N\_Nodes + N\_Relations)) \tag{3.14}$$

This approach to measuring clarity is simplistic, as the formula also suggests. However, its ease of use and intuition validate this implementation of the criterion.

The last criterion in the pragmatic layer is adaptability. We want this criterion to check whether the ontology has a grounded foundation and is easily extended with new information. A grounded foundation means that the information already present in the ontology is worked out well, and does not miss key concepts. This criterion is split in two. The first criterion is calculated by the ratio of leaf nodes to the amount of nodes within the ontology. The idea is that leaf nodes are positions in the ontology that can be extended easily, as they only have relations where they are the target but not the subject. Leaf nodes are defined as a node without outgoing arrows. See equation 3.15 for the formula.

$$Adaptability = (N\_Leafs/N\_Nodes + Avg\_Leaf\_Depth/Max\_Leaf\_Depth)/2 \tag{3.15}$$

The second criterion for adaptability looks at both the average and maximum leaf depth. If the the average leaf depth is close to the maximum leaf depth, we assure a grounded foundation of the ontology. The average of these two criteria together form the adaptability measure.

The pragmatic layer checks the ontology in the usability by measuring the clarity of the ontology, and it checks whether the ontology is easily extended with the adaptability criterion. These criteria fit in the semiotic framework, as well as in the literature about ontologies (Chapter 2).

3.3 Results

Our approach to ontology evaluation is based on metrics from the semiotic framework. This study is based on how humans interpret ontologies and how they evaluate them. Our work takes an ontology, and marks it with a number between 0 and 1. In order to find out whether our work is coherent with the way humans evaluate ontologies, we compare the fit of our work to the evaluation by humans.

For assessing the fit between our work and human evaluation, we used five ontologies from the BioPortal [31]. These ontologies were evaluated by humans in the work of McDaniel et al [21]. The human testers were given five ontologies, which they had to rate from a scale of 1 to 5 on five different attributes. They rated the ontologies based on the task of creating an ontology pertaining to blood. The results of our approach compared to these human testers are shown in table 3.2.

Ontology name	Human		Our work	
	Normalized	Rank	Normalized	Rank
NCIT - National Cancer Institute Thesaurus	.88	1	.54	3
UBER - Uberon Anatomy Ontology	.69	2	.49	5
ENVO – Environmental Ontology	.69	3	.51	4
FMA – Foundational Model of Anatomy	.64	4	.58	1
RADLEX – Radiology Lexicon	.32	5	.57	2

TABLE 3.2: Results

The normalized scores are a relative indication of how good an ontology is and therefore, the ordering of the whole set of ontologies is more expressive. A rank of 1 is the highest rank. We can immediately see that the fit between our work and the human evaluation is very low. We explain this by the fact that our work used more criteria than the five criteria used by the human testers. Therefore, our work is more strict on evaluating the ontologies than the human testers, leading to lower scores overall.

Since we designed our work to evaluate ontologies with the purpose of seeing how well on NLP-generated ontologies, we test this evaluation method by creating ontologies with three different tools: Stanford CoreNLP, co-occurrence and Hearst patterns. We asked three ontology experts to provide a scientific paper or article (i.e. a paper they wrote themselves) from which we would produce ontologies. The main advantage of asking ontology experts to provide feedback on the ontologies is that they know a lot about characteristics of good and bad ontologies. By giving them the power of choosing their own paper, we assured that they know the content of the paper and thus what should and should not be in the ontology.

The human evaluation and the evaluation from our work, is compared to the evaluation using the Doors framework. We do not only want to show that our work is fit to the task of evaluating NLP-generated ontologies, but we also want to show that our work is more precise in evaluating NLP-generated ontologies than the Doors framework. If it were so, there would have been no need for a new ontology evaluator.

The human testers were thus given three different ontologies based on the paper they provided. They could evaluate the ontologies on the same set of criteria (structure, richness, precision, accuracy, clarity and adaptability) rating each criterion on a Likert scale from 1 to 5, with 1 being the lowest possible score and 5 the highest.

NLP-tool	Human		Our work		Doors	
	Normalized	Rank	Normalized	Rank	Normalized	Rank
Stanford CoreNLP	.38	1	.60	1	.28	2
Co-occurrence	.29	2	.51	3	.27	3
Hearst Patterns	.27	3	.56	2	.30	1

TABLE 3.3: Results

We had created three different ontologies with each of the NLP-tools, based on three different papers. For the results as shown in 3.3, we took the mean of the evaluations over these by ontologies by the different ontology evaluators: the human testers, our work and Doors. We found that human testers were not pleased about the quality of the ontologies. Some statements on the different NLP-generated ontologies were: "this model is far from a complete reflection of the paper" and "this ontology missed the most important classes from the paper" . However, there were also some positive words about the ontologies and they were all consentient about the possibilities of the NLP-tools.

We see that human testers scored the ontologies by Stanford CoreNLP the best. This is in line with the evaluation of our work. The other two NLP-created ontologies scored similar, but definitely lower. Our work also evaluated them both lower than Stanford CoreNLP, but the second and third rank are swapped. This suggests that our work is fit for the task of evaluating NLP-generated ontologies.

Even though the absolute evaluations from Doors are close to the human evaluations, their relative ordering does not match the order of the humans. Stanford CoreNLP, is ranked second whilst Hearst patterns is ranked number one. Humans evaluated Hearst pattern ontologies as the worst ones. Therefore, we suggest that the Doors evaluations lack the insights in NLP-generated ontologies, which our work does seem to have.

The results also show that our work has higher absolute scores than the human testers. We address this to the fact that some criteria should be weighted more in the average than others. For example, most human testers said that they think that criteria from the semantic layer are more important that the syntactic layer. Also the clarity of an ontology was valued more than others.

3.4 Conclusion

Our approach is among the first automated ontology evaluators based on multiple criteria. Our work is similar to the evaluator by McDaniel et al [21] in the sense that both our approaches are based on the semiotic framework. Whereas their approach is simplistic in the sense that their criteria are calculated on simple graph measures, e.g. amount of nodes and leaf nodes, our approach builds upon this work by implementing more advanced criteria.

The use of more advanced criteria meant the algorithm gathered the results slower than the approach of McDaniel. This was the main downside when evaluating ontologies from the biology engineering world [31]. These ontologies are man-made,

have a complex, different namespace (no english words) and with the biggest one evaluated having over 150.000 nodes, they are relatively big.

Our evaluation framework had difficulties with evaluating the man-made ontologies from Bio-portal. This can be addressed to the fact that those ontologies had concepts in the form of URL's instead of English words. Our work proved to be more precise on ontologies that were created by NLP-tools, and even though our algorithm has a higher time-complexity, it is still useful as the task of evaluating ontologies does not have to be in real-time.

Another point we came across, is the human evaluation of NLP-generated ontologies. Humans tend to score these ontologies very low, and the remarks state that they are not useful nor complete. The criteria from our evaluator can be used to suggest improvements, as we will see in section 4.

Chapter 4

Improving ontologies

In the previous section we have seen an approach for automated ontology evaluation. The purpose for such an evaluation method is of course to be able to quickly determine from a set of ontologies, which among them is the best. On the other hand, in a world where ontologies can be made automatically, we want to be able to improve them automatically as well. An automated evaluator can be employed to use different techniques from the artificial intelligence domain. Such techniques can vary from neural networks to evolutionary computing. Due to the time constraints for this thesis, we limit our ontology improver to a simple heuristic as a proof of concept for the possibilities regarding an automated ontology evaluator.

This chapter is set up as follows. First, we will introduce one heuristic which adapts an ontology. Second, we will argue why this heuristic will improve ontologies and afterwards we can check the effects of applying this heuristic by passing it on to the ontology evaluator. Finally, we draw some conclusions from these results.

4.1 Ontology improvement heuristics

A heuristic is a simple rule that checks the ontology for some conditions, and if it encounters those, it will adapt the ontology in a predefined manner. Such a heuristic works if it is intuitive and simple. If it is not simple, it allows for too many errors to be made. An intuitive heuristic is easier to understand.

4.1.1 The heuristic

When we look at ontologies created by Stanford CoreNLP, we see a certain structure in the ontology which does not seem to be intuitive in relation to the input sentence. In figure 4.1 we see the same figure as in chapter 2 in the section on Stanford CoreNLP. The input sentence was: "The quick brown fox jumps over the lazy dog". We see in this figure that the concepts it extracts from this input sentence consist of: "fox", "brown fox", "quick brown fox", "dog" and "lazy dog". However, intuitively this does not seem be match our expectations. We see the concept "fox" occurring multiple times, with different adjectives. The same holds for the concept "dog".

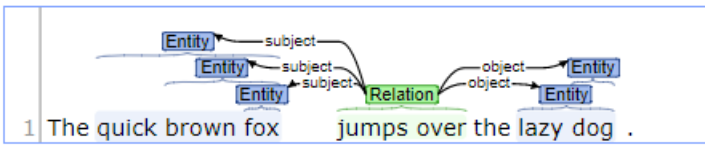

FIGURE 4.1: Snapshot from the online Stanford CoreNLP tool

Intuitively, we would like the concepts "fox" and "dog" to both occur only once. "Quick", "brown" and "lazy" would become new concepts which have a relation with

either "fox" or "dog". For example, "fox" could have a new relation "has colour" with the new concept "brown".

We implemented this simple yet intuitive idea as a heuristic to make small improvements over NLP-generated ontologies. The heuristic is defined as follows.

1. Iterate over every node n in nodes
2. If n consists of both adjectives and nouns (according to WordNet), go to 3, else go to 1
3. Split the adjectives from n and create a new node for every adjective. The noun(s) in n remain in the node and its relation with other nodes remain. Create a new relation between n and every new node. Furthermore, find every edge where n occurred in its initial state, and transform to its new state.
4. After iterating, merge duplicate nodes and edges, so that they all occur only once.

A sample node we could encounter in JSON format can be seen in figure 4.2. If the heuristic iterates over this node, it will find the structure of an adjective and a noun within one node. Notice that the order would not make a difference, nor would the addition of multiple adjectives or nouns in the identifier.


```
{
    "count": 1,
    "ID": "brown fox"
},
```


FIGURE 4.2: A sample node in JSON format

The heuristic will move on and split "brown" from "fox". This will create a new node, name "brown". The node in figure 4.2 is updated to be "fox". An edge is added between these nodes as can be seen in figure 4.3.


```
{
    "count": 1,
    "relation": "adjective",
    "source": "brown",
    "target": "fox"
},
```


FIGURE 4.3: A sample edge in JSON format

After these steps, every edge is updated where "brown fox" appeared as a source or a target of the relation. In the end all the nodes are merged, as during this heuristic it may be possible that a node is created that already exists.

4.1.2 Expectations

In this section we will review our heuristic in the picture of the framework behind the ontology evaluator. How do we expect this heuristic to change the evaluation of an ontology? The answer to this question helps us get a better understanding of the heuristic, but maybe even more importantly, it helps us get a better understanding of the evaluator itself. If a heuristic like this, that seems intuitive and solves a specific

aspect of NLP-generated ontologies, does not lead to a higher score by the evaluator, than it may be the case that our evaluator would be lacking in some aspects. We will review the heuristic by going through each layer and consider the effect of the changes on an ontology.

The heuristic makes a huge impact on a syntactic level. More nodes are created, and for each new node a new relation is created as well. In our approach these relations all will be the same, so the set of different relations can be increased by one at a maximum. We can conclude from this that the richness will go down a little because of this. We believe that this happens due to the fact that we name the relation "adjective" for all the adjectives extracted from the nodes. Since the newly created nodes always will be root nodes in the new ontology, we expect the amount of root nodes to go up as well. This means the score for the structure criterion of an ontology will slightly decrease as well.

In the semantic layer we had defined precision and accuracy. For precision it is easy to see that it can only go up. Nodes from the ontology that are mapped to keywords from a corpus will still map to those words, thus changing nothing. However, the new nodes may allow for more nodes to be mapped to new keywords. The variance will go up as well, but that might be harder to see.

The pragmatic layer is affected the most positive among the other layers. The clarity from the ontology will go up as the amount of nodes with adjectives or adverbs will go down. The amount of nodes will rise, thus leading to a higher clarity. The adaptability criterion will also go up, since the ontology will end up with more leaf nodes than before the heuristic. The reasoning behind this is the same as for the syntactic layer.

The only layer we expect to get a lower score is the syntactic layer. This can be partly solved by finding a way to define the relations between the new nodes and their origin node. However, we still expect the ontology to have a higher overall score because of the big changes in the pragmatic layer.

4.2 Results

Since we have an automated evaluator we can check the difference in evaluations of an ontology before and after the heuristics. We took one of the papers, also used for the evaluation process in the previous chapter. For each of the three different NLP-tools we use the ontology generated to be improved by the heuristic. Afterwards we pass this ontology trough to the evaluator again to check whether the heuristic has improved the quality measure given by the evaluator.

		Before		
NLP-tool	Syntactic	Semantic	Pragmatic	Quality
Stanford CoreNLP	.53	.48	.75	.59
Co-occurrence	.31	.50	.76	.52
Hearst Patterns	.20	.43	.72	.45
		After		
NLP-tool	Syntactic	Semantic	Pragmatic	Quality
Stanford CoreNLP	.53	.48	.71	.57
Co-occurrence	.31	.50	.77	.53
Hearst Patterns	.25	.43	.70	.46

TABLE 4.1: Results

In table 4.1 we see the evaluations from our evaluator before and after the heuristic was performed. We show the individual scores for the different layers from the semiotic framework to get more insight in the effect of the different criteria on the overall quality. We notice that the ontology from Stanford CoreNLP scored less after the heuristics. This is mainly due to the pragmatic layer. We stated that we expect the pragmatic layer to go up, in the previous subsection. However, we noticed that due to the amount of nodes being merged into one node, the amount of leaf nodes significantly dropped, leading to a lower score for adaptability. The change in clarity was not enough to compensate for this loss, leading to a lower score. For the ontologies made by the other two algorithms we do see a positive change in evaluation.

This is exactly the opposite of the effect we expected. We did not expect the adaptability criterion to drop substantially. The reason why it did is still not clear, so either the heuristic or the implementation is incorrect. Since the heuristic itself is intuitive, we think the heuristic is not the problem. Therefore, probably the implementation is not correct and should be fixed if it is so.

Our clarity measure went up only a little, but not always enough to compensate for the loss in adaptability. This could suggest two things. Either we should use a weighted evaluator, in which the weights of each criterion is different, or our clarity measure is not yet complete. We think both are true.

Firstly, the different criteria should have different weights. For example, when assessing a paper, both the format of the slides and the content are important. However, a mistake in the content of the presentation weighs more than a mistake in style of the slides. The same argument holds for ontologies. Certain criteria should weigh more than other.

Secondly, we think our criteria are not yet complete. For example, the clarity measure scores high, even though there are still many concepts present in the ontology that could be merged together. Since we only look at a specific kind of clarity, we have no way yet to automatically find other types of flaws in clarity. This is a flaw that potentially holds for all the criteria within our framework. However, we believe that the list of criteria and their implementation can reach an extensive point, as there is only a limited amount of criteria humans look at when evaluating an ontology.

4.3 Conclusion

We proposed a simple yet intuitive heuristic that should improve NLP-generated ontologies, since a quick look at these ontologies learnt us that they often have concepts that can be merged into a single, more specific concept. The results show that some of the ontologies get marginally better, but one of them also scored marginally lower. This has shown us that a simple modification can have a diverse impact on an evaluator based on multiple criteria.

Our heuristic was meant to show that a small progression in evaluation can be measured with the simplest rules. We believe that the possibilities are greater than such a simple heuristic. With the evolving field of data mining and natural language processing, there is no doubt that ontologies can be improved even further.

Chapter 5

Conclusion

We started this thesis with two main research questions. We try to answer those questions in this section, by taking in account the conclusions of the previous sections. We will end this final section with a small discussion of the shortcomings of our work, and how other researchers can build upon this work.

The first question was:

- What are quality measures humans use in their assessment of ontologies, and which of those are feasible to be implemented as quality measures for automatic ontology evaluation?

As we saw, there are a lot of similarities between semiotics and the way humans evaluate ontologies. An automated ontology evaluator should therefore implement the semiotic framework as well. We saw an implementation of this idea in the work of McDaniel et al [21]. However, we felt their implementation of criteria lacked a theoretical foundation on some points. We therefore built upon their idea, and added more criteria that we found during our literature study. The work done by McDaniel performed well on a set of man-made ontologies on a specific domain, but failed to show similar results on ontologies created by NLP-tools. Our work did not manage to perform accordingly on the man-made ontologies, but our approach performed better on the NLP-generated ontologies. We address this seeming paradox to two shortcomings of our work and that of McDaniel. Their work was limited by the implementation of the criteria. We tried to solve that by only implementing criteria that had enough foundations. However, doing so resulted in a limited amount of criteria. So in short, our work lacked in the extensiveness of criteria, where their work lacked in its implementation. However, we believe that our work is well-grounded and that it therefore is a good foundation for further extensions.

Our automated ontology evaluator was built with the idea of automatically improving NLP-generated ontologies. Our second research question was as follows:

- How can we improve existing NLP-tools by extending them with heuristic rules that implement ontology adjustments according to our quality measures?

An automated evaluation function gives access to different techniques of improving an ontology automatically. We have shown a heuristic of improving ontologies, based on a characteristic that NLP-generated ontologies possess.

Both parts of our work had its shortcomings. The evaluator should be extended to include more of the criteria humans use in their evaluation of ontologies. However, as we have seen, the implementation of new criteria should be theoretically grounded to make sure the quality of the evaluation stays as close to that of humans. As we have also seen, more research should be done towards a weighted

evaluation function, as some criteria should weigh more than others. Our heuristic did not have that much impact, as it only made small modifications to the ontology.

A suggestion for future work would also be to look into combining different aspects of the collection of available NLP-tools. With a multiple criteria based evaluator we can find the strengths and weaknesses of each tool. Combining the strengths of the different tools could possibly lead to a better ontology than those created by a single tool.

Bibliography

[1] Vrandečić, D. (2009). Ontology evaluation. In Handbook on Ontologies (pp. 293-313). Springer Berlin Heidelberg.

[2] Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2006, June). Modelling ontology evaluation and validation. In ESWC (Vol. 4011, pp. 140-154).

[3] Hlomani, H. (2014). Multidimensional Data-driven Ontology Evaluation (Doctoral dissertation).

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

[5] Brewster, C., Alani, H., Dasmahapatra, S., & Wilks, Y. (2004). Data driven ontology evaluation.

[6] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., & Shadbolt, N. R. (2003). Automatic ontology-based knowledge extraction from web documents. IEEE Intelligent Systems, 18(1), 14-21.

[7] Völker, J., Vrandečić, D., & Sure, Y. (2005, November). Automatic evaluation of ontologies (AEON). In International Semantic Web Conference (pp. 716-731). Springer, Berlin, Heidelberg.

[8] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220.

[9] Brank, J., Grobelnik, M., & Mladenić, D. (2005). A survey of ontology evaluation techniques.

[10] Li, Z., Raskin, V., & Ramani, K. (2007). A methodology of engineering ontology development for information retrieval. In Proceedings of the 16th International Conference on Engineering Design (ICED'07.

[11] Ta, C. D., & Thi, T. P. (2015, November). Automatic Evaluation of the Computing Domain Ontology. In International Conference on Future Data and Security Engineering (pp. 285-295). Springer, Cham.

[12] Dellschaft, K., & Staab, S. (2006, November). On how to perform a gold standard based evaluation of ontology learning. In International Semantic Web Conference (Vol. 4273, pp. 228-241).

[13] Guarino, N., Oberle, D., & Staab, S. (2009). What is an Ontology?. In Handbook on ontologies (pp. 1-17). Springer Berlin Heidelberg.

[14] Summit, O. (2013). Communique. Towards Ontology Evaluation across the Life Cycle.

[15] Smith, B., & Welty, C. (2001, October). Ontology: Towards a new synthesis. In Formal Ontology in Information Systems (Vol. 10, No. 3, pp. 3-9). ACM Press, USA, pp. iii-x.

[16] Pulido, J. R. G., Ruiz, M. A. G., Herrera, R., Cabello, E., Legrand, S., & Elliman, D. (2006). Ontology languages for the semantic web: A never completely updated review. Knowledge-Based Systems, 19(7), 489-497.

[17] Cali, A., Gottlob, G., & Pieris, A. (2012). Towards more expressive ontology languages: The query answering problem. Artificial Intelligence, 193, 87-128.

[18] Prud, E., & Seaborne, A. (2006). SPARQL query language for RDF.

[19] Porzel, R., & Malaka, R. (2004, August). A task-based approach for ontology evaluation. In ECAI Workshop on Ontology Learning and Population, Valencia, Spain (pp. 1-6).

[20] Hofmann, T. (1999, July). Probabilistic latent semantic analysis. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 289-296). Morgan Kaufmann Publishers Inc.

[21] McDaniel, M., Storey, V. C., & Sugumaran, V. (2018). Assessing the quality of domain ontologies: Metrics and an automated ranking system. Data & Knowledge Engineering.

[22] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

[23] Cimiano, P., Pivk, A., Schmidt-Thieme, L., & Staab, S. (2005). Learning taxonomic relations from heterogeneous sources of evidence. Ontology Learning from Text: Methods, evaluation and applications.

[24] Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, 13(01), 157-169.

[25] Socher, R., Bengio, Y., & Manning, C. (2013). Deep learning for NLP. Tutorial at Association of Computational Logistics (ACL), 2012, and North American Chapter of the Association of Computational Linguistics (NAACL).

[26] Sánchez, D., Batet, M., Martínez, S., & Domingo-Ferrer, J. (2015). Semantic variance: an intuitive measure for ontology accuracy evaluation. Engineering Applications of Artificial Intelligence, 39, 89-99.

[27] Stamper, R. (1991). The semiotic framework for information systems research. Information systems research: Contemporary approaches and emergent traditions, 515-528.

[28] Alexopoulos, P., & Mylonas, P. (2014, May). Towards vagueness-oriented quality assessment of ontologies. In Hellenic Conference on Artificial Intelligence (pp. 448-453). Springer, Cham.

[29] Fernández, M., Overbeeke, C., Sabou, M., & Motta, E. (2009, December). What makes a good ontology? A case-study in fine-grained knowledge reuse. In Asian Semantic Web Conference (pp. 61-75). Springer, Berlin, Heidelberg.

[30] Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing.

[31] Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., & Musen, M. A. (2011). BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic acids research, 39(suppl_2), W541-W545.

[32] Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.