Pattern-Based Ontology Construction from Selected Wikipedia Pages

Carmen Klaussner
University of Nancy 2
carmen@wordsmith.de

Desislava Zhekova
University of Bremen
zhekova@uni-bremen.de

Abstract

In this paper, we describe how ontologies can be built automatically from definitions obtained by searching *Wikipedia* for lexico-syntactic patterns based on the hyponymy relation. First, we describe how definitions are retrieved and processed while taking into account both recall and precision. Further, concentrating only on precision, we show how a consistent and useful domain ontology can be created with a beneficial precision of 80%.

1 Introduction

Knowledge bases are created to depict models of the world in the way we perceive it (Lacy, 2005). Nowadays, the general concern about the representation and communication of information increases the need to do the latter in a more meaningful and structured way (Brachman, 1983). Natural Language Processing (NLP) is a task, which is relatively easy for humans, but presents a complex computational challenge, as machines need carefully structured and well-designed content to unambiguously interpret information (Lacy, 2005). Ideally, one creates hand-crafted thesauri, such as *WordNet*[1], which are more reliable, but with information constantly changing, their coverage falls behind and costs of maintenance remain high. Thus, the possibility of creating knowledge bases from regularly updated knowledge sources, such as *Wikipedia*[2], which offers a vast amount of information on a wide variety of topics, seems to be a desirable solution for this difficult situation.

In this paper, we present the *Ontology creator (Oc)*[3], which extracts articles from *Wikipedia*, searches them for definitions and transfers the results into an appropriate knowledge representation using the ontology language *OWL*[4]. For this purpose, we use lexico-syntactic patterns that were reported to enable successful extraction of semantic relations (Hearst, 1992; Hearst, 1998; Mititelu, 2006; Mititelu, 2008). We evaluate the overall system performance and concentrate on successful hyponymy patterns in order to improve the resulting ontology's precision. Our hypothesis is that, by searching *Wikipedia* for the hyponymy relation, one can create consistent domain ontologies that can be easily used as good knowledge bases.

Thus, section 2 gives an overview of related projects. In section 3 we introduce the *Ontology creator* and describe how patterns are built and represented in the knowledge base. Further, in section 4 we evaluate the system performance and describe the most common errors that we observed. Section 5 closes with a concluding comment.

2 Related Work

In order to be able to extract definitions from domain-independent, unrestricted text, methods for discovering lexico-syntactic patterns are generally used, employing English corpora (Hearst, 1992; Hearst, 1998; Mititelu, 2006; Mititelu, 2008), such as the British National Corpus. Lexico-syntactic patterns can model semantic relations, such as hyponymy (the notion of hyponym–hypernym in the sense that if L_0 is a (kind of) L_1, then L_1 is hypernym to L_0 (Hearst, 1992)). As reported by Mititelu (2008), some patterns' success rates reach 100%. Suchanek et al. (2008) also used *Wikipedia* as the information base. The authors extract facts from *Wikipedia*'s infoboxes and combine these with the category structure of *WordNet* into an ontology. In this way, they maintain a clearly-structured hierarchy of word senses, enriched by *Wikipedia*'s vast amount of information with a final precision of 95%.

[1] http://wordnet.princeton.edu/
[2] http://en.wikipedia.org/wiki/
[3] http://sourceforge.net/projects/ontocreation/
[4] http://www.w3.org/TR/owl-ref/

Proceedings of the Student Research Workshop associated with RANLP 2011, pages 103–108,
Hissar, Bulgaria, 13 September 2011.

3 Ontology Creation

In order to examine our hypothesis, we designed a system, the *Ontology creator*, that extracts definition relations from *Wikipedia* and converts them into a representation in *OWL*. Thus, in section 3.1 we introduce the system module that collects the articles. Further, in section 3.2, we introduce the parser that is used to assign grammatical structure to the individual sentences. Section 3.3 focuses on the lexico-syntactic patterns and section 3.4 explains how a pattern match is represented in *OWL*.

3.1 Extracting from Wikipedia

For the purpose of building a domain-specific ontology that concentrates on only one area of knowledge, it is necessary to collect articles that are highly topically-interlinked. Consequently, for the acquisition of articles from *Wikipedia*, we use a webcrawler, that starts with a given article and collects pages that have a referring link to it. We employ the open-source webcrawler *JSpider*[5], which is a highly configurable Web Spider engine. It allows to limit the search to only one website, to set the depth into its structure as well as the MIME type and to restrict the number of resources to be fetched per site. These features are all important to keep the articles' topics as closely related as possible. Currently, the depth level is set to two, which will produce a fair number of connected pages.

3.2 Parsing Articles

To gain a more accurate basis for the pattern search, the *Oc* uses the *Stanford parser* (Klein and Manning, 2003) to derive grammatical structures for each sentence. To bridge the stages from the HTML article to a usable list representation for the parser, we used the *DocumentPreprocessor*[6].

3.3 Building Lexico-Syntactic Patterns

For extracting definitions from text, we make use of lexico-syntactic patterns indicative of the hyponymy relation. Since definitions represent statements about the world, they are often expressed in terms of each other, where one concept is used to define another one (Brachman, 1983). Hyponymy, or the *IS-A* link, is one of the most basic types of conceptual relations for categorising classes of things in the world represented, carrying with it the notion of an explicit taxonomic hierarchy (Brachman, 1983). One deterministic characteristic of a taxonomic hierarchy is that all members inherit the properties of their respective superclass by virtue of being an instance of that class (*inheritance of properties*) (Brachman, 1983). Classes can be made up of subclasses or individuals. Classes may be viewed as classifying types, since they are abstract concepts of physical or virtual objects in the world. If a class is a subclass to another one, it will introduce a more specific concept than its superclass. Members of a class are instantiations of a particular class concept.

(1) *An apple is a fruit.*

Example (1) is the explicit version of ordinary hyponymy, which allows the inheritance of lexical semantic properties. There are lexico-syntactic patterns for different semantic relations, although hyponymy seems to yield the most accurate results. Yet, in order to use a specific pattern, one has to define how its variables are realised in natural language (exemplified in (2)) (Hearst, 1992):

(2) $NP_0\ such\ as\ NP_1, NP_2..., (and \mid or) NP_n$
$for\ all\ NP_i,\ 1 \leq i \leq n, hyponym(NP_i, NP_0)$

Building a search pattern for the above example is realised when an NP_0 (indicating the superclass) is represented by a single noun phrase consisting of a proper noun or a determiner, a noun and an optional adverbial phrase, whereas $NP_1, NP_2..., (and \mid or) NP_n$ may consist of more than one of the above noun phrases. Using these specifications to search for definitions in the sentence: *"Other forms of deception, such as disguises or forgeries, are generally not considered lies, though the underlying intent may be the same."*, one obtain matches as: *hyponym("forgery","deception")*, *hyponym("disguise","deception")*.

The patterns that were integrated into the *Oc* (listed in table 1) were suggested by Hearst (1992) and extended by Mititelu (2008). We used a subset of them, consisting of those rated the highest (discarding patterns for lack of results or for performance reasons). We also modify pattern 11 to admit plural matches and synonyms. In order to optimise the pattern search, we use the *JRegex*[7] li-

[5] http://j-spider.sourceforge.net/
[6] http://www.koders.com/java/
[7] http://sourceforge.net/projects/jregex/files/

No.	Pattern
1.	NP_0 *including* NP_{1+i}
2.	NP_0 *such as* NP_{1+i}
3.	*by such* NP_0 *as* NP_{1+i}
4.	NP_0 *(mainly | mostly | notably | particularly | usually | especially | principally)* NP_{1+i}
5.	NP_0 *in particular* NP_{1+i}
6.	NP_0 *like | except* NP_{1+i}
7.	NP_0 *for example* NP_{1+i}
8	NP_0 *other than* NP_{1+i}
9.	*state(=*NP_0*) of *ment(=*NP_{1+i}*)*
10.	NP_0 *i.e. | e.g.* NP_{1+i}
11.	NP_0*, (a) kind(s) | type(s) | form(s) of* NP_{1+i}

Table 1: Patterns for the acquisition of definitions

brary as well as *Commons Lang*[8].

3.4 Data Processing

In order to depict obtained definitions, we use an ontology representation. In the context of computer and information sciences, ontologies are meant to formally describe the terminological concepts and relationships that constitute a domain and generally provide a more common understanding of it as well as one that can be communicated between humans and machines. Thus, an ontology is a formally described, machine-readable collection or vocabulary of terms and their relationships and is used for knowledge sharing and reuse. Ontologies are encoded into files using ontology languages. A taxonomical ontology is the most common form of an ontology. It consists of a hierarchy of concepts which are related with specialisation *IS-A* relationships (Lacy, 2005). *OWL* is one of the languages that can be used to define ontologies and the associated individual data. For this project, we use the *OWL DL* dialect, as it supports consistency checks and reasoning and thus allows us to infer new facts from existing ones. The hyponymy relation in *OWL* can be expressed through the use of the relation between a superclass and its subclasses or members. Since we have only general indications of what the various matches can look like, we use a processing approach that is appropriate for most entities. The first decision to be taken is whether to make a noun phrase into a new class or an individual. This, however, is only relevant for NP_{1+i} since NP_0 always has instances and therefore always constitutes a class. An individual is only created if all its substrings have been classified as proper nouns by the *Stanford parser*, otherwise the current string is processed as a subclass. All modifiers are set to become subclasses of the predefined `characteristicValues` class and are linked to the respective class through the `hasCharacteristic` property. The number of superclasses/subclasses in a match is also dependent on the number of modifiers of both NP_0 and NP_{1+i}. If we consider as an example the following: *"... primitive animals, such as starfish ..."* that leads to the relation: *hyponym("starfish"-*NP_1*,"primitive animal"-*NP_0*)*, where a modifier of NP_0 is present , there will be a class `Animal`, which will be superclass to `PrimitiveAnimal` in an intersection with `hasCharacteristic some Primitive` and `Animal`. This in turn will be superclass to the `Starfish` class. We assume that nouns that are modified by some adjective would constitute an own concept and will only be more specific through this addition.

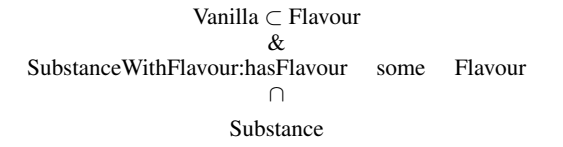

Figure 1: Complex subclass example in OWL

Superclasses that consist of multiple nouns will not be subdivided any further, since one cannot assume that each noun by itself will actually constitute an own class or convey a separate concept in the same way. Figure 1 depicts the conversion of NP_0 featuring a head with an of-complement as in *"... flavour of substance, such as vanilla,..."*. In this case `Flavour` is made into a class with subclasses: `Vanilla,...etc.` and linked through a new property called `hasFlavour`, which has `Flavour` as its range, to the new `SubstanceWithFlavour` class, which will be subclass to a general `Substance` class.

This representation may not always be the most suitable, but concepts introduced by an of-complement[9] do present a difficult case. The processing of NP_{1+i} featuring an of-complement cannot be done in the same way, since there is no range for a possible property relation as in the previous example. Furthermore, the concept introduced by it is usually already rather specific. Although *OWL* allows defining distinct members and disjoint classes to mark mutual distinctness, we cannot in general make all classes or individuals

[8] http://commons.apache.org/lang/

[9] Apart from "of", "for" and "in" were also allowed in the relation.

relations	count	detailed	
matched	65	ideal	52
		incomplete	7
		parser error	6
not matched	122	missing pattern	73
		parser error	2
		ambiguity	47
total	187		

Table 2: System results.

Precision	0.80
Recall	0.32
F_1-Measure	0.46
$F_{0.5}$-Measure	0.62

Table 3: Precision, recall and f-measures of *Oc*.

of a match mutually distinct/disjoint, since tests showed that two names, for instance, in an enumeration sometimes refer to the same individual.

4 Experiments

In order to investigate the system performance in regard to recall and precision, in section 4.1, we look at the performance overall and in section 4.2, we attempt to fine-tune the system to obtain the highest possible results in regard to precision.

4.1 System Evaluation

For the purpose of evaluating *Oc*'s performance, its final output was compared to a gold-standard. Therefore, we let the program process 20 topically-related *Wikipedia* article extracts (a total of 641 sentences) and compare the results to the gold-standard analysis of the same sentences. The manual analysis was subject to various criteria. A definition relation was only recognised as relevant or correct if the subclass/individual clearly translated into a sub-concept/instance of the superclass. Moreover, it was also evaluated how useful or appropriate the match is. We do not count more complex concepts that consist of a head with more than one of-complement or relations where subclass and superclass are separated by extra nested relative clauses. A match, whether successful or not, consists of a superclass and its subclass/individual. Relations that are obtained by simple derivation of the system are not counted, for example *"... snacks such as nuts, dried fruit, ..."* results in: *hyponym("Nut","Snack")*, *hyponym("DriedFruit","Snack")* and *hyponym("DriedFruit","Fruit")*. Yet, the third relation is derivative from the entity itself and is therefore not counted.

Table 2 lists all matched relations and the ones that are appropriate but were not matched. Of 187 relations in the sample, 65 were captured and 122 were not retrieved. Further, we show detailed distribution of all matched relations, of which 52 were matched ideally, 7 incompletely (parts were missing) and 6 incorrectly. We also show the various categories of relations that were not found. 73 are not retrieved because there is no appropriate capturing pattern yet, 2 are due to incorrectly-assigned parser tags and 47 matches are not found because of missing patterns. Thus, the *Oc* manages to reach a recall of 32% and precision of 80%. The F_1 measure with recall and precision weighed equally lies at 46%. Yet, using an abundant source, such as *Wikipedia*, takes the burden from the general lack of data, which allows us to rate precision twice as high as recall. Thus, $F_{0.5}$ marks a 62% overall system performance. A more systematic representation of the figures is shown in table 3.

4.1.1 Reasons for Non-Retrieval

Table 2 divides not matched relations into different categories:

Missing Patterns: If we consider the sentence: *"Piquance is considered another such basic taste in the East."*, we see that a pattern, such as $NP_1\ VP*\ another\ such\ NP_0$ is needed. Similarly, in 73 of the 122 cases where relations are not captured a new pattern can be added.

Ambigious Patterns: Further, in 39% of the cases the appropriate pattern was also missing, but not as easy to replace as in the aforementioned scenario. For example in: *"Couverture is a term used for chocolates rich in cocoa butter..."* the range, *"term used for chocolates rich in cocoa butter"*, is a complex concept that is difficult to convert into one distinct superclass. An additional problem is that the mechanism would go as far as *"term"* and then stop, resulting only in Couverture $\subset$ Term. Although it is technically possible to check that there is no identifying clause following the prospective superclass, this has proved even for smaller cases to be extremely time-consuming and since the classic *IS-A* pattern did not account for many cases, it seemed wiser to forgo this option and leave out the pattern entirely. An even more difficult matter is presented by *"The word cacao itself derives from the* `Nahuatl, Aztec`

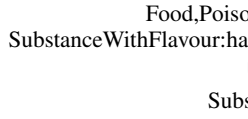

Figure 2: False processing example

category	count	%
Matched Relations	1706	100%
Correct Relations	1389	81%
Incorrect Relations	317	19%

Table 4: Ontology evaluation system results.

`language`, *word cacahuatl...”* leading to *hyponym(“Nahuatl”,“Aztec language”)*. There is little in regard to distinguishing environment to seperate results of mere enumerations from one of two terms that are in a hyponymy relationship and separated by a comma. Such examples lead to a drastic reduction in precision. Thus, it seemed sensible to process more articles on the topic to compensate for potential matches of these cases.

4.1.2 Captured Relations Issues

Having described frequent issues connected to definitions that were not captured, we now examine the ones that were retrieved. Even though with 80% the overall precision is reasonably satisfactory, we now also consider whether the respectively assigned representation in *OWL* is appropriate.

Patterns not Exclusive to Hyponymy: In some cases incorrect processing is due to the fact that the pattern is not exclusive to hyponymy, but covers simple non-hyponymy sentences as well, as in *“The majority of the Mesoamerican people made chocolate beverages, including Aztecs, who made...”* Yet, there are patterns that are more reliable to produce hyponymy and for the benefit of higher precision, one can concentrate on those.

Incomplete: The question of the appropriate representation in *OWL* is more difficult to evaluate. Most issues concern heads with an of-complement as superclass, where it is not clear what the subsequent clause is referring to. In most cases the *OWL* results are not wrong, but in at least 2 cases they seem awkward. For example: *“It refers to the ability to detect the flavour of substances such as food, certain minerals, and poisons, etc.”* results in the structure in figure 2, while it should have processed as Food, Poison... ⊂ Substance. This is partly due to the ambiguity resulting from the scope of the noun phrase referenced, for which the parser did not make any difference in structure. Thus, it is worthwhile to reconsider the way the of-complement is processed.

Creating an ontology from a small set of sentences, as was done here, is bound not to yield a large ontology. Our system obtained a total of 52 correct relations from searching 641 sentences. Most relations are topically-related, although there are some which are not, since referring links sometimes bear only a remote relation. In this ontology, which started on the term “chocolate”, we also find facts about dialectologists. Creating larger ontologies may circumvent such problems which *Wikipedia*’s ample resources would also allow. Yet, what is essential and of primary interest to us, is the ontology’s correctness and appropriateness.

4.2 Ontology Evaluation

After both a quantitative and qualitative analysis of the *Oc*, which were able to highlight the more frequent issues in connection to pattern-based ontology construction, we now concentrate on further enhancing precision. Recall can be increased by adding new patterns or widen the scope of the existing ones, although in our case it is essential not to compromise precision in any way. In this context, we choose to give precision a clear priority, since we are not looking for as many relations as possible, but for as many correct ones as possible. In order to further precision we concentrate on more successful hyponymy patterns and use a larger sample to obtain more accurate results. The articles were collected across a couple of different topics to also test for the patterns’ suitability independent of the genre. Since this is a larger sample, we are not able to make a close analysis of whether the match yielding a relation is appropriate, as we had done during the first test. Important is only whether the final relation in the ontology is correct and appropriate in terms of content, superclass/subclass relation and processing, as we aim to determine the usefulness of the ontology overall. For the current experiment, we retained patterns: 2, 3, 4, 5, 8 and 9 (table 1). Table 4 depicts the final system results from it. The first row displays the number of retrieved relations overall (1706), followed by the number of correct ones (81%) and incorrect ones

Tomato $\subset$ Fruit
Tomato $\subset$ Vegetable

Figure 3: Contradictory facts

(19%). The of-complement matches appeared in 6% of all matched relations. The final system precision based on the ontology evaluation is slightly higher than the precision we achieved by evaluating the general system performance, which is not surprising since building larger ontologies also leads to a more accurate evaluation as well as to more overlaps of facts, where incorrect ones sometimes compromise correct ones. In the following part, we look at the different issues the resulting ontology poses and how they can be addressed in future research.

General Issues One of the general issues that can be observed is a classification problem. A specific entity is sometimes classified as two slightly controversial things. The facts in figure 3 both appear in the ontology, resulting from a more biological classification of tomato (fruit) and a maybe slightly more practical one (vegetable). Both facts are correct for their scope, but e.g. make it impossible to have fruit and vegetable as disjoint classes. At the moment, there is no component that deals with this issue, as a relative correctness would maybe also depend on the application area. Issues with of-complements, as have already been described in 4.1, remain. In the current set, 6% of the matches had an of-complement. In general, the easiest option is to disregard matches with these grammatical structures completely, however, if genuine they do express a particular kind of relationship that would not be captured in quite the same way otherwise.

Necessary Additions Until now, we had not included past participle in verb phrases. However, there are examples which show that this decision should be re-assessed: *"...NP JJ alcoholic NNS beverages, RB especially NP VBN distilled NNS beverages..."*. Since "distilled" is disregarded, this yields the relations *hyponym("alcoholic beverage","beverage")*, which is derived from the superclass: "alcoholic" modifying "beverage", and *hyponym("beverage","alcoholic beverage")* and consequently, every "beverage" is also per definition an "alcoholic beverage".

In addition, it would be beneficial to create more disjunct members or disjoint classes which would provide a possibility for a self-check in the *Oc*. Although, as has already been pointed out, classification issues may render this difficult.

5 Conclusion and Future Work

The overall aim of this project is to build consistent domain ontologies from facts obtained from websources, such as *Wikipedia*. In our work, a large number of relations remained unmatched, but the high precision encourages further research in this area. Since *Wikipedia* is an extremely big resource, one can afford losing prospective facts for the benefit of obtaining a more correct and hence more useful knowledge base. In order to enhance the *Oc*, it would be necessary to put more work towards the idea of disjointness and disjunct members in *OWL*. This way the resulting ontology would be more alert to irregularities. Further, one could evaluate the different patterns in regard to their respective ambiguity dimensions.

References

Ronald J. Brachman. 1983. What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks. *Computer*, 16(10):30–36, Oct.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA. ACL.

Marti A. Hearst. 1998. Automated discovery of WordNet relations. In *C. Fellbaum, WordNet: An Electronic Lexical Database*, pages 131–153. MIT Press.

Dan Klein and Christopher D. Manning. 2003. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Lee W. Lacy. 2005. *Owl: Representing Information Using the Web Ontology Language*. Trafford Pub.

Verginica B. Mititelu. 2006. Automatic Extraction of Patterns Displaying Hyponym-Hypernym Co-Occurence from Corpora. In *Proceedings of CESCL.*

Verginica Barbu Mititelu. 2008. Hyponymy Patterns. In *Proceedings of the 11th international conference on Text, Speech and Dialogue*, TSD '08, pages 37–44, Berlin, Heidelberg. Springer-Verlag.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. YAGO: A Large Ontology from Wikipedia and . *J. Web Sem.*