A Hybrid Approach for Learning SNOMED CT Definitions from Text

Felix Distel and Yue Ma*

Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany,
{felix,mayue}@tcs.inf.tu-dresden.de

Abstract. In recent years approaches for extracting formal definitions from natural language have been developed. These approaches typically use methods from natural language processing, such as relation extraction or syntax parsing. They make only limited use of description logic reasoning. We propose a hybrid approach combining natural language processing methods and description logic reasoning. In a first step description candidates are obtained using a natural language processing method. Description logic reasoning is used in a post-processing step to select good quality candidate definitions. We identify the corresponding reasoning problem and examine its complexity.

1 Introduction

Throughout the medical domain the formal representation of knowledge has proven to be beneficial, allowing for powerful reasoning services for the debugging and querying of knowledge bases. Among these KR formalisms lightweight Description Logics (DL) from the $\mathcal{EL}$ family such as OWL2EL [12] have proven to be especially successful: they allow for tractable reasoning while still providing a level of expressivity that is sufficient for most ontologies in the medical domain. One such ontology is SNOMED which is now a widely accepted international standard [15].

The downside of formal semantics is the cost associated with creating, maintaining and extending ontologies. Since the knowledge is represented using logic based syntax and semantics these tasks require specially trained staff that are both experts in logics and in the application domain. One approach to facilitate the work of these experts is to mine the vast expanse of knowledge that is available in textual form, e.g. in PubMed, textbooks or on the web. A number of natural language based approaches have been proposed to automatically or semi-automatically convert concept descriptions that are available in textual form into formal concept descriptions in DL. The descriptions are obtained through analysis of lexical or linguistic features, without a mechanism that checks if the resulting descriptions are logically sound.

Typically, one sentence in natural language describes only one aspect of a target concept, it hardly ever provides a full definition. To obtain full definitions the partial definitions from different sentences must be compared and the ones with highest quality must be selected. Typically, the NLP formalism will provide a confidence value for each sentence which can provide some indication of its quality.

* Funded by the DFG Research Unit FOR 1513, project B1.

Logic based knowledge representation is not very robust when it comes to errors in concept descriptions. Even less obvious errors can cause an ontology to yield unwanted consequences or even to become inconsistent. In this work we propose to use formal constraints on the target concept in order to ensure at least a certain degree of logical soundness.

The proposed approach is therefore a hybrid approach. In a first step an NLP formalism is used to obtain candidate definitions from text and in a second step a reasoning problem is used in order to select a good subset of the candidate definitions. In Section 5 we provide a summary of our text mining algorithm which was first presented in [10, 16]. It uses a multi-class classifier to predict if a given sentence describes an existential restriction and if so, for which role. In Section 6 the task of selecting good candidates is formalized as a reasoning problem. The proposed formalization extends an idea from [9]: the conjunction over the selected set of candidates should satisfy the constraints while maximizing the accumulated confidence values. In [9] the minimum has been used to accumulate confidence values, whereas here, we investigate how the choice of accumulation function influences the complexity of the reasoning problem (Section 7).

2 Related Work

For a long time, algorithms for learning ontologies from natural language have focussed mainly on learning subclass relationships [20, 14], in a linguistic context sometimes called hyponyms [5]. These algorithms essentially learn a taxonomy at best.

The generation of complex terminological axioms using more advanced logical constructors is clearly a much harder task [4, 19]. There are essentially two underlying ideas for the generation of complex axioms. In works such as [18, 19] the syntax of the natural language input sentences is parsed, i.e. sentences are broken down into functional units. These syntax trees are then scanned for predefined patterns which correspond to a certain type of logical constructor. An interactive approach for ensuring the correctness of the mined axioms is presented in [13]. The rules transforming lexical and linguistic patterns to logical syntax are typically manually created [17]. This makes it difficult to adapt these approaches to new domains as extensive tweaking of the rules is required.

By contrast approaches based on machine learning techniques are easier to adapt. Some of these [8, 3] work on an instance level and are typically based on Inductive Logic Programming techniques. On the terminological level there are approaches based on relation extraction techniques [11]. Here, the patterns themselves are learned from annotated text. In scenarios where the set of role names is stable, these techniques can be used to learn simple restrictions of role depth 1 [10, 16].

3 Preliminaries

We consider the lightweight Description Logic $\mathcal{EL}$, whose *concept descriptions* are built from a set of concept names N_C and a set of role names N_R using the constructors *top concept* $\top$, *conjunction* $\sqcap$, and *existential restrictions* $\exists$. The semantics of $\mathcal{EL}$ is defined using interpretations $\mathcal{I} = (\Delta_\mathcal{I}, \cdot^\mathcal{I})$ consisting of a non-empty *domain* $\Delta_\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$ mapping role names to binary relations on $\Delta_\mathcal{I}$ and concept descriptions to subsets of $\Delta_{\mathcal{I}}$ according to Table 1. A concept description C is said to be atomic if $C \in N_C \cup \{\top\}$ or $C = \exists r.D$ for some $r \in N_R$ and some concept description D.

Table 1. Syntax and Semantics of $\mathcal{EL}$

Name	Syntax	Semantics
concept name	A	$A^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}}$
role name	r	$r^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$
top concept	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{x \mid \exists y\colon (x,y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
primitive definition	$A \sqsubseteq C$	$A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
full definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$

As *axioms* we allow full definitions and primitive definitions. *Full definitions* are statements of the form $A \equiv C$, *primitive definitions* are statements of the form $A \sqsubseteq C$ where A is a concept name and C is a concept description. A TBox $\mathcal{T}$ is a set of axioms of these two types. We say that the interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ if $A^{\mathcal{I}} = C^{\mathcal{I}}$ (or $A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$) holds for every full definition $A \equiv C$ (primitive definition $A \sqsubseteq C$. respectively) from $\mathcal{T}$. A concept description C is said to be subsumed by the concept D with respect to the TBox $\mathcal{T}$ (denoted by $\mathcal{T} \models C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{T}$. It is well-known that subsumption reasoning in $\mathcal{EL}$ is tractable, i.e. given concept descriptions C and D, and a TBox $\mathcal{T}$ it can be decided in polynomial time if $\mathcal{T} \models C \sqsubseteq D$ [2].

There are two reasons for our restriction to full definitions and primitive definitions instead of the more expressive GCIs. In our chosen setting we try to learn concept descriptions for SNOMED CT, which uses only full definitions and primitive definitions. Second, this restriction allows us to check subsumption on an atomic level according to the following lemma from [9].

Lemma 1 ([9]). *Let $\mathcal{T}$ be a TBox containing only primitive definitions. Let C and D be concept descriptions that can be written as $C = C_1 \sqcap \cdots \sqcap C_n$ and $D = D_1 \sqcap \cdots \sqcap D_m$, where C_i, D_j are atoms for all $1 \leq i \leq n$, $1 \leq j \leq m$. Then $\mathcal{T} \models C \sqsubseteq D$ iff for every atom D_i, $1 \leq i \leq m$, there is an atom C_j, $1 \leq j \leq n$, such that $\mathcal{T} \models C_j \sqsubseteq D_i$.*

4 Task Description

Our approach for learning SNOMED CT-descriptions from text is based on the following observations. When comparing legacy versions of SNOMED CT to the current one, it is obvious that the set of role names has remained relatively stable while the number of concepts has increased. In our approach we therefore assume the set of role names to be fixed.

In our approach the knowledge engineer is allowed to pick a target concept, i.e. a concept name that is not already defined in SNOMED CT. Our goal is, by benefiting from DL reasoning, to facilitate the knowledge engineer's design task by automatically generating a definition for the target concept from the natural language text corpus.

We proceeds in two steps. First, in the text mining step the text corpus is searched for sentences containing the target concept. A relation mining algorithm then decides for each of these sentences whether it describes a primitive definition of the target concept. For example the sentence "Baritosis is pneumoconiosis caused by barium dust." describes the primitive definition

$$\mathsf{Baritosis} \sqsubseteq \exists \mathsf{causative\_agent.Barium\_Dust}.$$

The right hand side of the primitive definition, in this case $\exists \mathsf{causative\_agent.Barium\_Dust}$ is then added to the set of candidate descriptions. For each candidate description, the text mining algorithm also provides a numerical value, indicating confidence in the correctness of the candidate.

Second, for the reasoning step, a set of formal constraint is obtained beforehand, e.g. from the SNOMED CT design manual [6]. The objective is to select a subset of the set of candidate description that

- satisfies the constraints, and
- maximizes an accumulation function over the confidence values.

5 Text Mining Step

The text mining step has been previously presented in [10, 16]. We thus only provide a summary of the approach here. The main idea is to use existing SNOMED CT descriptions to train a multiclass classifier to recognize sentences describing existential restrictions.

Data Preparation During the data preparation phase each sentence must be scanned for occurrences of SNOMED CT concepts and these concepts must be mapped to the corresponding part of the sentence. State of the art annotators are available to perform this task. In [10] the tool *MetaMap* [1] is used, and in [16] *MetaMap* is combined with a purpose built annotator.

We use existing knowledge from SNOMED CT to create our training set. In order to make use of implicit knowledge, the set

$$\mathcal{R} = \{(A, r, B) \mid A, B \in N_C, r \in N_R, \text{SNOMED CT} \models A \sqsubseteq \exists r.B\}$$

is considered. Now, if a sentence is annotated with both A and B for some triple (A, r, B) in SNOMED CT, then the sentence is labelled with the class r. That is, perhaps a bit naively, whenever two concepts A and B occur in the same sentence and SNOMED CT entails the subsumption $A \sqsubseteq \exists r.B$ then in the training phase the sentence is assumed to describe this relation. An example for the data preparation of a sentence is shown in the first two lines of Table 2.

Table 2. Text Alignment and Features

Annotated Sentence	"*Baritosis*/Baritosis_(disorder) is pneumoconiosis caused by *barium dust*/Barium_Dust_(substance)."		
SNOMED CT relationship	Baritosis_(disorder) | Causative_agent | Barium_Dust_(substance)		
Features	left type	between-words	right type
	disorder	"is pneumoconiosis caused by"	*substance*
BoW	{is, pneumoconiosis, caused, by}		
Word 2-grams	{is, pneumoconiosis, caused, by, is pneumoconiosis, pneumoconiosis caused, caused by }		
Char. 2-grams	{i, s, p, n, e, u, m, o, c, a, d, b, y, is, pn, ne, eu, um, mo, oc, co, on, ni, io, os, si, ca, au, us, se, ed, by}		

Training Phase To train a multiclass classifier features need to be extracted from each of those sentences that have been assigned a class in the data preparation step. In [10] in addition to the between words (words occurring between the parts of the sentence annotated with the two SNOMED CT-concepts) the types of the two SNOMED CT concepts are considered. The between words are represented as character n-grams. In [16] only the between words are considered and a comparison between representations as word n-grams, character n-grams and bag of words is made.

Test Phase Based on the weights learned in the training phase, the multiclass classifier can predict for new annotated sentences whether they describe an existential restriction. These existential restriction are then added to the candidate set. Typically, the classifier will also give a confidence value, indicating how likely it assumes the prediction to be correct. The exact nature of the confidence value depends on the classifier used.

An experimental evaluation of the approach can be found in both [10] and [16]. Due to limited availability of high quality text the evaluations consider only the three roles causative_agent, associated_morphology, and finding_site. These roles have been selected because they are relatively frequently used in SNOMED CT and occur in the same fragment of SNOMED CT, the fragment describing diseases. In [10] the influence of the quality of the text corpus is examined by comparing text from Wikipedia to text obtained using the tool Dog4Dag [20]. It is shown that the latter, which is less noisy, yields a visible increase in the quality of the extracted $\mathcal{EL}$ descriptions. In [16] a comparison between different state of the art supervised learning algorithms is made: logistic regression, support vector machines, multinomial naive Bayes, and random forests. It appears that support vector machines provide the best quality results, reaching an f-measure of up to 82.9% for the role causative_agent.

Both evaluations indicate that while the quality of the results is decent, especially considering the relatively naive generation of the training set, there is still room for improvement. We propose to use DL reasoning in a post-processing step to improve the results.

6 Reasoning Step

In this section we formally define the reasoning problem used to select a good set of candidate descriptions. We assume that a set of constraints can be obtained as described Section 6.1. In our setting constraints are simply GCIs or negated GCIs where either the left-hand side or the right-hand side is a concept variable X. We distinguish constraints of the following four types:

$$D \sqsubseteq X \qquad (1) \qquad\qquad D \not\sqsubseteq X \qquad (3)$$

$$X \sqsubseteq D \qquad (2) \qquad\qquad X \not\sqsubseteq D \qquad (4)$$

In these constraints D can be a complex concept description. X must be a concept name not occurring in $\mathcal{T}$, D, or another constraint. For a complex concept description C in which no concept variables occur, we say that C satisfies the positive constraint $D \sqsubseteq X$ or $X \sqsubseteq D$ of X if $\mathcal{T} \models D \sqsubseteq C$ or $\mathcal{T} \models C \sqsubseteq D$, respectively. C satisfies the negative constraint $D \not\sqsubseteq X$ or $X \not\sqsubseteq D$ if $\mathcal{T} \not\models D \sqsubseteq C$ or $\mathcal{T} \not\models C \sqsubseteq D$, respectively.

In its simplest form the task is now straightforward: for a given set of description candidates and a given set of constraints, find a subset of the candidates whose conjunction satisfies the constraints. For complexity considerations we restate this as a decision problem.

Problem 1 (Concept Selection (CS)). Input: A set of atomic candidate concept descriptions $\mathcal{S}$, a (possibly empty) ontology $\mathcal{T}$ and a set of constraints $\mathcal{C}$ of the forms (1)–(4).
Question: Is there a subset $\mathcal{S}'$ of $\mathcal{S}$ such that $\bigsqcap \mathcal{S}'$ satisfies all the constraints in $\mathcal{C}$?

In the text mining step, each candidate in $\mathcal{S}$ has been assigned a confidence value between 0 and 1. In practice one would like to give preference to solutions of CS that contain candidates that have been assigned higher confidence values in the text mining approach. We therefore need to decide upon a function for accumulating the confidence values of all the selected candidates. Intuitively, such a function should be associative, commutative, non-decreasing and have 1 as unit, in other words it should be a triangular norm or t-norm [7]. The most well-known t-norms are

- the *minimum t-norm*, also known as *Gödel t-norm*, $x \otimes y = \min(x, y)$,
- the *product t-norm*, $x \otimes y = x \cdot y$, and
- the *Łukasiewicz t-norm*, $x \otimes y = \max\{0, x + y - 1\}$.

In the following we shall only consider t-norms whose computation takes only polynomial time in the size of the inputs x and y. The idea behind the following decision problem is to maximize the accumulated confidence of the selected candidates with respect to a given t-norm $\otimes$.

Problem 2 (Accumulated Confidence Concept Selection ($\otimes$-CS)). Input: A set of atomic candidate concept descriptions $\mathcal{S}$, a real number $m \in [0, 1]$, a (possibly empty) ontology

$\mathcal{T}$ and a set of constraints $\mathcal{C}$ of the forms (1)–(4), together with a confidence function $\mathrm{wt}\colon \mathcal{C} \to [0, 1]$.

Question: Is there a subset $\mathcal{S}'$ of $\mathcal{S}$ such that $\bigsqcap \mathcal{S}'$ satisfies all the constraints in $\mathcal{C}$ and $\bigotimes\{\mathrm{wt}(S) \mid S \in \mathcal{S}'\} \geq m$?

While the minimum t-norm typically has the best computational properties, it can in practice make sense to use one of the other two t-norms. Consider a situation where several candidates have the same confidence value $\mathrm{wt}(C) = c < 1$. For the minimum t-norm it makes no change whether the selection contains one or many of these candidates, while the other two t-norms give preference to smaller selections, thereby likely increasing the precision of the solution.

6.1 Obtaining the Constraints

In an ideal scenario constraints of types (1)–(4) should come directly from the knowledge engineers. Since this, however, would simply shift the cost from the creation of concept descriptions to the creation of constraints, other ways of obtaining the constraints need to be examined.

In the case of SNOMED CT, a surprisingly large number of constraints can be found in its design manual, the SNOMED CT User Guide [6]. For instance, it restricts the range of the role finding_site to Body_Structure. One can thus add a restriction $X \not\sqsubseteq \exists \mathsf{finding\_site}.C$ for all those concepts describing disjoint main branches to Body_Structure, such as Disorder, Substance, etc. This ensures that the learned concept does not become unsatisfiable. A similar approach can be used for domain restrictions.

Furthermore, a great number of definitions in SNOMED CT are only partial definitions. If the task is to enrich this partial definition using text mining, then the existing definition can be used as type (2) constraints.

Finally, one could also consider an interactive way for obtaining constraints. In a first approximation, one would simply use the conjunction over the complete set of candidates as the definition of the target concept. This definition would be temporarily added to the ontology. Both the original and the extended ontology would then be classified. The user is then presented with a set of subsumptions between concept names that are entailed by the extended ontology but not by the original one. He can then choose to mark each of them as *intended* or *unintended*. The former are added as positive constraints, while the latter are added as negative constraints. The concept selection problem is solved again. The entire process is repeated with the new selection of candidates until the user no longer marks any new consequences as unintended.

Example 1. Take Baritosis as the target concept. In an experiment described in [9], the following description candidates were extracted with high weights:

$\exists$causative_agent.Barium_Compound,	0.92140
$\exists$causative_agent.Dust,	0.97038
$\exists$finding_site.Lung_structure,	0.99999
$\exists$causative_agent.Barium_Dust,	0.99997

Due to the high weights returned by the learning approach, it is impossible to exclude any of the candidates based on weights alone. However, the concept selection framework can discard the incorrect candidate $\exists$Causative_agent.Barium_compound, since the SNOMED CT User Guide states that the range of causative_agent is restricted to exclude Chemical, translating to the constraint

$$X \not\sqsubseteq \exists \mathsf{causative\_agent}.\mathsf{Chemical}.$$

Barium_Compound which is subsumed by Chemical can thus be discarded.

7 Complexity

In [9] we have looked at the complexity of CS and $\otimes$-CS but only for the case where $\otimes$ is the minimum t-norm (called $\min$-CS in the following). The results from [9] state that both CS and $\min$-CS are NP-complete, but become tractable when only constraints of types (1)–(3) are allowed. NP-hardness is thus caused by constraints of type (4).

In Section 7.1, we argue that for $\min$-CS tractability is still maintained if constraints of all four types are allowed but we require that in every type (4) constraint $X \not\sqsubseteq D$ the concept D must be atomic. Conversely, in Section 7.2 we show that for the product and Łukasiewicz t-norm $\otimes$-CS is NP-complete even if only (2) constraints are used.

7.1 Tractable Variants

In [9] it is shown that CS and $\min$-CS are tractable when only constraint types (2) to (3) are used. In this section we recall the argument and show that it can easily be extended to constraints of type

$$X \not\sqsubseteq D, \text{ where } D \text{ is atomic,} \qquad (4')$$

using Lemma 1. Throughout this subsection we assume that full definitions have previously been expanded and the TBox contains only primitive definitions. The reason is, that in the presence of full definitions one could simply add a full definition for D to the TBox and thus emulate type (4) restrictions using type (4') restrictions.

We first consider the variant that restricts to constraint types (2) and (3).

Restriction to (2) and (3): First, let an instance $(\mathcal{T}, \mathcal{S}, \mathcal{C})$ of CS be given. Notice that if a concept C satisfies a constraint $X \sqsubseteq D$ or $D \not\sqsubseteq X$ and E is a concept description satisfying $E \sqsubseteq C$ then E also satisfies the constraint. In particular, if $\bigsqcap \mathcal{S}'$ satisfies all constraints for some $\mathcal{S}' \subseteq \mathcal{S}$ then $\bigsqcap \mathcal{S}$ also satisfies them. Hence, if only constraint types (2) and (3) occur, then there is a solution to the CS problem iff $\mathcal{S}$ itself is a solution. The latter can be verified in polynomial time since subsumption reasoning in $\mathcal{EL}$ is tractable. The same argument shows that there is a solution to the $\min$-CS problem $(\mathcal{T}, \mathcal{S}, k, \mathcal{C}, \mathrm{wt})$ iff $\{S \in \mathcal{S} \mid \mathrm{wt}(S) \geq k\}$ is a solution. Again, this can be verified in polynomial time. Hence, CS and $\min$-CS are tractable if we restrict to types (2) and (3).

Restriction to (1)–(3) + (4'): We now assume $\mathcal{T}$ contains primitive definitions only, i.e. all full definitions have been expanded. Consider now a constraint $D \sqsubseteq X \in \mathcal{C}$ of type (1). A concept $\bigsqcap \mathcal{S}'$ for $\mathcal{S}' \subseteq \mathcal{S}$ satisfies this constraint if and only if $\mathcal{T} \models D \sqsubseteq S$ for all $S \in \mathcal{S}'$.

For a constraint $X \not\sqsubseteq D \in \mathcal{C}$ of type (4') we can use Lemma 1 to exploit the fact that D is atomic and that $\mathcal{T}$ only contains primitive definitions. The Lemma then states that a set $\mathcal{S}' \subseteq \mathcal{S}$ satisfies $\mathcal{T} \models \bigsqcap \mathcal{S}' \sqsubseteq D$ iff there is some $S \in \mathcal{S}'$ such that $\mathcal{T} \models S \sqsubseteq D$. In other words $\mathcal{S}'$ satisfies the constraint $X \not\sqsubseteq D \in \mathcal{C}$ iff $\mathcal{T} \not\models S \sqsubseteq D$ for all $S \in \mathcal{S}'$.

This shows that there is a solution $\mathcal{S}'$ of $(\mathcal{T}, \mathcal{S}, \mathcal{C})$ iff there is a solution $\mathcal{S}_0'$ of $(\mathcal{T}, \mathcal{S}_0, \mathcal{C}_0)$ where

$$\begin{aligned} \mathcal{S}_0 &= \{S \in \mathcal{S} \mid \forall (D \sqsubseteq X) \in \mathcal{C} \colon \mathcal{T} \models D \sqsubseteq S\} \\ &\quad \cap \{S \in \mathcal{S} \mid \forall (X \not\sqsubseteq D) \in \mathcal{C} \colon \mathcal{T} \not\models S \sqsubseteq D\} \\ \mathcal{C}_0 &= \{c \in \mathcal{C} \mid c \text{ of type (2) or (3)}\}. \end{aligned} \tag{5}$$

Notice that $\mathcal{C}_0$ can be obtained in linear time and $\mathcal{S}_0$ can be computed in polynomial time since subsumption reasoning in $\mathcal{EL}$ is tractable. This shows that restrictions of type (1) can be dealt with in a polynomial time preprocessing step. Tractability of CS and $\min$-CS for constraints of types (1)–(3) then follows immediately from this fact and tractability of CS and $\min$-CS when restricted to (2) and (3).

This shows that while constraints of type (4') may still be problematic in the context of full definitions, tractability is maintained for TBoxes where the expansion of full definitions does not lead to an exponential blowup.

7.2 NP-hard variants

For all variants of CS and $\otimes$-CS containment in NP is easy to see, since one can simply guess a subset of $\mathcal{S}$ and verify in polynomial time if it is a solution (remember that subsumption reasoning in $\mathcal{EL}$ is tractable). Notice, that in $\otimes$-CS we require that the t-norm itself can be computed in polynomial time.

In [9] it is shown using a reduction from SAT that CS with all 4 types of constraints is NP-hard. This implicitly proves NP-hardness for $\otimes$-CS, since CS can be considered to be a special case of $\otimes$-CS with all confidence values set to 1.

In this work, using a reduction from the well-known NP-complete problem Vertex Cover, we show that for the product and Łukasiewicz t-norm $\otimes$-CS is NP-hard, even when restricted to constraints of type (2).

Problem 3 (Vertex Cover). Input: A natural number k and an undirected graph $G = (V, E)$ consisting of a set of vertices V and a set of edges E.

Question: Is there a subset $O \subseteq V$ satisfying $O \cap e \neq \emptyset$ for every edge $e \in E$ and $|O| \leq k$.

We describe the reduction for the product t-norm. Starting with an instance of Vertex Cover consisting of a graph G and a number k, we construct an instance of $\otimes$-CS as follows. We choose $N_R = \{r\}$, $N_C = \{A_e \mid e \in E\}$. For every node $v \in V$ we introduce a candidate concept description

$$S_v = \exists r. \bigsqcap_{v \in e} A_e. \tag{6}$$

The confidence function wt is set to be constantly 0.5 (any value strictly between 0 and 1 serves the purpose). For every edge $e \in E$ a constraint

$$X \sqsubseteq \exists r.A_e \tag{7}$$

is added to the set of constraints. We then define $m = 0.5^k$ and we let the TBox $\mathcal{T}$ be empty.

Consider a subset $\mathcal{S}' \subseteq \{S_v \mid v \in V\}$. Since the TBox $\mathcal{T}$ is empty the conjunction $\bigsqcap \mathcal{S}'$ satisfies the constraint $X \sqsubseteq \exists r.A_e$ iff $S_v \sqsubseteq \exists r.A_e$ holds. From (6) this is equivalent to $v \in e$. This shows that $\bigsqcap \mathcal{S}'$ satisfies all constraints iff $\{v \mid S_v \in \mathcal{S}'\} \cap e \neq \emptyset$ for all $e \in E$, i.e. iff $\{v \mid S_v \in \mathcal{S}'\}$ is a Vertex Cover of G. Furthermore, since wt is constantly 0.5, we have

$$m = 0.5^k \leq \bigotimes \{\mathrm{wt}(S_v) \mid S_v \in \mathcal{S}'\} = 0.5^{|\mathcal{S}'|} \tag{8}$$

iff $k \geq |\mathcal{S}'| = |\{v \mid S_v \in \mathcal{S}'\}|$. This shows that $\{v \mid S_v \in \mathcal{S}'\}$ is a solution cover to the given instance of Vertex Cover iff $\mathcal{S}'$ is a solution to the constructed instance of $\otimes$-CS. Since Vertex Cover is known to be NP-complete this shows that $\otimes$-CS is NP-hard. Since we already know that it is contained in NP we obtain NP-completeness.

Lemma 2. *If $\otimes$ is the product t-norm then $\otimes$-CS is NP-hard, even when restricted to constraints of type (2).*

In the case of the Łukasiewicz t-norm nearly the same reduction can be used. The only modification that needs to be made is that wt should to be constantly $\frac{k}{k+1}$ and $m = \frac{1}{k+1}$. Then again

$$\begin{aligned} m = \frac{1}{k+1} \leq \bigotimes \{\mathrm{wt}(S_v) \mid S_v \in \mathcal{S}'\} &= \max\{0, \frac{k}{k+1} \cdot |\mathcal{S}'| - (|\mathcal{S}'| - 1)\} \\ &= \max\{0, 1 - \frac{1}{k+1} |\mathcal{S}'|\} \end{aligned} \tag{9}$$

holds iff $k \geq |\mathcal{S}'| = |\{v \mid S_v \in \mathcal{S}'\}|$.

Lemma 3. *If $\otimes$ is the Łukasiewicz t-norm then $\otimes$-CS is NP-hard, even when restricted to constraints of type (2).*

The complexity results are summarized in Table 3.

8 Conclusion and Future Work

In this paper we have extended the framework from [9] for obtaining $\mathcal{EL}$-concept descriptions from text in natural language. The framework proposes a hybrid approach where in a first text mining step description candidates are mined, and good candidates

Table 3. Summary of Complexity Results

	without full definitions			with full definitions		
	(1)–(3)	(1)–(3) + (4')	(1)–(4)	(1)–(3)	(1)–(3) + (4')	(1)–(4)
CS, min-CS	P	P	NP	P	NP	NP
Ł-CS, Π-CS	NP	NP	NP	NP	NP	NP

are selected in the reasoning step by solving a constraint problem. We have theoretically analyzed the complexity of the concept selection problem $\otimes$-CS. If the minimum t-norm is used, the problem remains tractable for a restricted type of constraints. We have further shown that its complexity increases from P to NP when the product or Łukasiewicz t-norm are used instead of the minimum. Regardless of the accumulation function, the problem is intractable for the full set of constraints.

In this paper, we have not presented an experimental evaluation. However, in an earlier work such an evaluation has been performed for the case of the minimum t-norm [9]. In this evaluation, a text corpus obtained from the web was used. The experiment was restricted to frequent roles from the disease branch of of SNOMED CT. Concepts were removed and then relearned from the text corpus using the presented approach. The hybrid approach showed significant improvements when constraints were used over a pure text mining approach: for most concepts the precision increased to 100%.

In the present work, the definition of the target concept was always obtained as a conjunction over the selected candidates. This is a good approach when the candidates are too general to describe the concept. It is, however, also possible that a candidate is too specific. In this case it could make sense to consider the least common subsumer (lcs) of the selected candidates, in order to generalize. For the least common subsumer we do not expect the concept selection problem to be tractable, or even in NP, since even the size of the lcs of a set of concept descriptions can be exponential in the size of the set.

We also plan to investigate, how well the approach performs for ontologies other than SNOMED CT. For the approach to be applicable, these ontologies need to satisfy a number of requirements. For the text mining step it is necessary that

- the set of roles is small and stable, and
- there is already a large set of concept definitions available, which can be used to train a classifier.

For the reasoning step a source for the constraints, such as a design manual, is required.

Finally, since for many classifiers the confidence values returned by the text mining step are probabilistic in nature, one might investigate the use of probabilistic extensions to $\mathcal{EL}$.

References

1. A. R. Aronson and F.-M. Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of IJCAI'05*, 2005.
3. M. Chitsaz, K. Wang, M. Blumenstein, and G. Qi. Concept learning for $\mathcal{EL}$++ by refinement and reinforcement. In *Proceedings of PRICAI'12*, pages 15–26, 2012.
4. P. Cimiano. *Ontology learning and population from text - algorithms, evaluation and applications*. Springer, 2006.
5. M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of Coling'92*, pages 539–545. Association for Computational Linguistics, 1992.
6. International Health Terminology Standards Development Organisation. SNOMED CT user guide. `http://www.snomed.org/ug.pdf`, 2013.
7. E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Springer-Verlag, 2000.
8. J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1-2):203–250, 2010.
9. Y. Ma and F. Distel. Concept adjustment for description logics. In *Proceedings of K-Cap'13*, 2013. to appear.
10. Y. Ma and F. Distel. Learning formal definitions for Snomed CT from text. In *Proceedings of AIME'13*, 2013. to appear.
11. M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL/AFNLP'09*, pages 1003–1011, 2009.
12. B. Motik, P. F. Patel-Schneider, and B. Parsia. OWL 2 web ontology language structural specification and functional style syntax. W3C Recommendation, October 2009.
13. S. Rudolph, J. Völker, and P. Hitzler. Supporting lexical ontology learning by relational exploration. In *Conceptual Structures: Knowledge Architectures for Smart Applications*, pages 488–491. Springer, 2007.
14. D. Sánchez and A. Moreno. Automatic generation of taxonomies from the www. In *Practical Aspects of Knowledge Management*, pages 208–219. Springer, 2004.
15. SNOMED *Clinical Terms*. Northfield, IL: College of American Pathologists, 2006.
16. G. Tsatsaronis, A. Petrova, M. Kissa, Y. Ma, F. Distel, F. Baader, and M. Schroeder. Learning formal definitions for biomedical concepts. In *Proceedings of OWLED'13*, 2013. to appear.
17. L. Vasto Terrientes, A. Moreno, and D. Sánchez. Discovery of relation axioms from the web. In *Knowledge Science, Engineering and Management*, volume 6291 of *Lecture Notes in Computer Science*, pages 222–233. Springer, 2010.
18. J. Völker. *Learning expressive ontologies*. PhD thesis, Universität Karlsruhe, 2009.
19. J. Völker, P. Haase, and P. Hitzler. Learning expressive ontologies. In *Proceeding of OLP'08*, pages 45–69, 2008.
20. T. Wächter, G. Fabian, and M. Schroeder. DOG4DAG: semi-automated ontology generation in OBO-Edit and protégé. In *Proceedings of SWAT4LS'11*, pages 119–120, 2011.