ELASTIQ: Answering Similarity-threshold Instance Queries in $\mathcal{EL}$

Andreas Ecke[1⋆], Maximilian Pensel[1⋆⋆], and Anni-Yasmin Turhan[2⋆⋆]

[1] Institute for Theoretical Computer Science,
Technische Universität Dresden
[2] Department of Computer Science, University of Oxford, UK

Abstract. Recently an approach has been devised how to employ concept similarity measures (CSMs) for relaxing instance queries over $\mathcal{EL}$ ontologies in a controlled way. The approach relies on similarity measures between pointed interpretations to yield CSMs with certain properties. We report in this paper on ELASTIQ, which is a first implementation of this approach and propose initial optimizations for this novel inference. We also provide a first evaluation of ELASTIQ on the GeneOntology.

1 Introduction

Description Logics (DLs) are a family of knowledge representation whose formal semantics allow the definition of a variety of reasoning services. The most prominent ones are *subsumption* and *instance query answering*. However, many applications need to query the knowledge base in a more relaxed manner. For instance, in the application of service matching TBoxes are employed to describe types of services. Here, a user request for a service specifies several requirements for the desired service and can be represented by a complex concept. For such a concept the ABox that contains the individual services is searched for a service matching the specification by performing instance query answering. In cases where an exact match with the provided requirements is not possible, a 'feasible' alternative should be retrieved from the ABox.

To relax the notion of instance query answering one can simply employ fuzzy DLs and perform query answering on a fuzzy variant of the initial query concept. However, on the one hand reasoning in fuzzy DLs easily becomes undecidable, see [2–4] and on the other hand fuzzy concepts would always relax the initial concept in a uniform way and cannot consider user or request-specific preferences on which parts of the query are more important and should not be relaxed.

A reasoning service that allows for a given query concept the selective and gradual extension of the answer set of individuals is answering of relaxed instance queries, was proposed in [7] and further investigated in [8, 6, 9]. The selective, gradual relaxation of the answer sets returned to instance queries is achieved by the use of concept similarity measures. A *concept similarity measure* (CSM)

⋆ Supported by DFG Graduiertenkolleg 1763 (QuantLA).
⋆⋆ Partially supported by DFG in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing".

yields, for a pair of concepts, a value from the interval $[0, 1]$—indicating how similar the concepts are. To answer a relaxed instance query is to compute for a given concept C, a CSM $\sim$, and a threshold t between 0 and 1, a set of concepts such that each of these concepts are similar to C by a threshold of at least t, if measured by the CSM $\sim$, and then finding all their instances.

Concept similarity measures are widely used in ontology-based applications. For ontologies from the bio-medical field, such as the GeneOntology ontology [10], they are employed to discover functional similarities of genes. Furthermore, CSMs are used in ontology alignment algorithms. For DLs there exists a whole range of CSMs, which could be employed for the task of answering relaxed instance queries [1, 5, 12, 15]. In particular the CSMs generated by the framework described in [12] allow users to specify which part of the ontology's signature is to be regarded more important when it comes to the assessment of similarity of concepts. Thus, these measures naturally allow users to select important features of the query concept and which aspect of the query concept to relax.

We investigated algorithms for computing answers to relaxed instance queries for $\mathcal{EL}$. This DL has good computational properties and large, well-known biomedical ontologies such as the GeneOntology [10] are written in (polynomial extensions of) $\mathcal{EL}$. Our algorithm for computing relaxed instances w.r.t. $\mathcal{EL}$-KBs with general TBoxes relies on canonical models of the query concept and of the queried KB. The employed CSM is derived from a similarity measure for pointed interpretations, which essentially implements relaxed forms of equisimulation between interpretations. A similar idea in spirit is pursued in [16, 17] for $\mathcal{EL}$-concepts, where similarity is measured in terms of 'how much is missing' to establish a homomorphism between graph representations of $\mathcal{EL}$-concepts.

Now, for computing answers to relaxed instance queries, the similarity values for all pairs of elements between the two canonical models need to be computed in the worst case. Thus a naive implementation would hardly be efficient. We report in this paper in first optimizations for this novel inference, which we have implemented in the system ELASTIQ ($\mathcal{EL}$ answering of similarity-threshold instance queries). A first evaluation on the GeneOntology shows that the proposed optimizations are vital for this kind of inference, but that response times for a single query over large ontologies are still about a second.

The remainder of the paper in structured as follows. Next, we give an introduction to the technical terms used and the relaxed instance inference. In Section 3 we discuss the algorithm for computing relaxed instances and the similarity measures employed for it. The ELASTIQ reasoner is introduced in Section 4 together with some optimizations and the evaluation of its performance on the GeneOntology. The paper ends with conclusions and future work.

2 Preliminaries

$\mathcal{EL}$-concepts are built from two mutually disjoint sets N_C of *concept names*, and N_R of *role names* using the syntactic rule: $C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C$, where $A \in N_C$ and $r \in N_R$. The semantics of $\mathcal{EL}$-concepts are defined by means

of interpretations, in which concept names are interpreted as subsets of the interpretation domain and roles as binary relations. The semantics are extended to complex concepts as usual. An $\mathcal{EL}$-*TBox* consists of a finite set of *general concept inclusion axioms* (GCIs) of the form $C \sqsubseteq D$. An interpretation is a *model* for a TBox if it satisfies all its GCIs. An $\mathcal{EL}$-*ABox* describes individuals from a set N_I of *individual names* using concept assertions of the form $C(a)$ and role assertions of the form $r(a, b)$. Again, an interpretation is a model for an ABox if it satisfies all its assertions. An $\mathcal{EL}$-*knowledge base* (KB) is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

The following commonly used reasoning tasks are implemented in most DL reasoning systems. *Concept subsumption* $C \sqsubseteq_{\mathcal{T}} D$ asks, for a TBox $\mathcal{T}$ and two concepts C and D, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. Given an individual a, a concept C, and a KB $\mathcal{K}$, a is called an *instance of* C w.r.t. $\mathcal{K}$, denoted $\mathcal{K} \models C(a)$, iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a concept C, *instance retrieval* returns all individuals from $\mathcal{A}$ that are instances of C.

For the DL $\mathcal{EL}$, the polynomial-time complexity of most reasoning procedures rely on the fact that canonical models can be built, from which it is possible to read off entailments directly. These canonical models represent the most general model for a concept or the individuals of an ABox w.r.t. to a TBox. Before we can formally define these canonical models, we need to introduce some notation. If X is a concept, TBox, ABox, or KB, then:

- $\mathrm{Sig}(X)$ denotes the signature of X; that is, the set of concept, role, and individual names appearing in X, and
- $\mathrm{sub}(X)$ is the set of all sub-concepts of concepts occurring in X.

Definition 1. *(canonical models) Let C be an $\mathcal{EL}$-concept and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an $\mathcal{EL}$-KB. The* canonical model $\mathcal{I}_{C,\mathcal{T}} = (\Delta^{\mathcal{I}_{C,\mathcal{T}}}, \cdot^{\mathcal{I}_{C,\mathcal{T}}})$ of C *w.r.t. the TBox $\mathcal{T}$ is:*

- $\Delta^{\mathcal{I}_{C,\mathcal{T}}} = \{d_C\} \cup \{d_D \mid \exists r.D \in sub(C) \cup sub(\mathcal{T})\}$
- $A^{\mathcal{I}_{C,\mathcal{T}}} = \{d_D \mid D \sqsubseteq_{\mathcal{T}} A\}$, *for all concept names A, and*
- $r^{\mathcal{I}_{C,\mathcal{T}}} = \{(d_D, d_E) \mid D \sqsubseteq_{\mathcal{T}} \exists r.E\}$ *for all role names r.*

The canonical model $\mathcal{I}_{\mathcal{K}} = (\Delta^{\mathcal{I}_{\mathcal{K}}}, \cdot^{\mathcal{I}_{\mathcal{K}}})$ of the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *is defined as follows:*

- $\Delta^{\mathcal{I}_{\mathcal{K}}} = \{d_a \mid a \in Sig(\mathcal{A}) \cap N_I\} \cup \{d_C \mid \exists r.C \in sub(\mathcal{A}) \cup sub(\mathcal{T})\}$,
- $A^{\mathcal{I}_{\mathcal{K}}} = \{d_D \mid D \sqsubseteq_{\mathcal{T}} A\} \cup \{d_a \mid \mathcal{K} \models A(a)\}$,
- $r^{\mathcal{I}_{\mathcal{K}}} = \{(d_D, d_E) \mid D \sqsubseteq_{\mathcal{T}} \exists r.E\} \cup \{(d_a, d_D) \mid \mathcal{K} \models \exists r.D(a)\} \cup \{(d_a, d_b) \mid r(a, b) \in \mathcal{A}\}$.

Note that canonical models for $\mathcal{EL}$ are always finite. Canonical models can be used to decide instance checking problems since a is an instance of C w.r.t. a KB $\mathcal{K}$ if and only if $a^{\mathcal{I}_{\mathcal{K}}}$ is an element of $C^{\mathcal{I}_{\mathcal{K}}}$ in the canonical model $\mathcal{I}_{\mathcal{K}}$ [13].These canonical models and their use for instance checking, are important for the algorithm for answering relaxed instance queries in Section 3.

Instance checking can be relaxed by using a concept similarity measure. Such a measure $\sim$ is a function that assigns to each pair of concepts (w.r.t. a TBox $\mathcal{T}$) a similarity value from the interval $[0, 1]$ with $C \sim C = 1$ for all concepts C. A

value $C \sim D = 0$ means that the concepts C and D are totally dissimilar, while a value of 1 indicates total similarity. A set of properties for CSMs was presented in [12]. In particular, a CSM $\sim$ is called *symmetric*, iff $C \sim D = D \sim C$; *equivalence invariant*, iff for all $C \equiv_{\mathcal{T}} D$ and all concepts E it holds that $C \sim E = D \sim E$; and *equivalence closed*, iff $C \equiv_{\mathcal{T}} D \iff C \sim D = 1$. Using those similarity measures, we define relaxed instances as follows:

Definition 2 (relaxed instance). *The individual a is a* relaxed instance *of the query concept Q w.r.t. the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the CSM $\sim_{\mathcal{T}}$ and the threshold $t \in [0, 1)$ iff there exists a concept X such that $Q \sim_{\mathcal{T}} X > t$ and $\mathcal{K} \models X(a)$.*

To compute the relaxed instances of an $\mathcal{EL}$-concept (w.r.t. an $\mathcal{EL}$-KB) it is not feasible to compute all sufficiently similar concepts and then perform instance checking for those, since (1) the number of such concepts can be infinite leading to an infinite number of queries and (2) a CSM does not necessarily provide a method how to obtain a 'sufficiently similar' concept.

3 The Algorithm for Computing Relaxed Instances

In [9], we proposed the CSM $\sim_c$ to be used to answer relaxed instance queries. This measure compares two concepts C and D w.r.t. a TBox $\mathcal{T}$ by computing their canonical models $\mathcal{I}_{C,\mathcal{T}}$ and $\mathcal{I}_{D,\mathcal{T}}$ and comparing the structure of the models starting from the elements d_C and d_D, respectively. For this, we define a similarity measure $\sim_i$ on pointed interpretations. Given a pointed interpretation $p = (\mathcal{I}, d)$, we denote with

- $\mathrm{CN}(p) = \{A \in N_C \mid d \in A^{\mathcal{I}}\}$ the set of concept names that d is an instance of in $\mathcal{I}$, and
- $\mathrm{SC}(p) = \{(r, (\mathcal{I}, e)) \mid (d, e) \in r^{\mathcal{I}}\}$ the set of direct successors of d in $\mathcal{I}$.

The interpretation similarity $\sim_i$ to be defined depends on three parameters:

1. A *primitive measure* $\sim_p : N_C \times N_C \cup N_R \times N_R \to [0, 1]$ assigns a similarity value to each pair of concept names and each pair of role names. Any primitive measure has to satisfy that $x \sim_p x = 1$ for any concept or role name x. Additionally, for the similarity measure $\sim_i$ to be symmetric, $\sim_p$ needs to be symmetric as well. We give a default primitive measure, that simply assigns similarity 0 to pairs of different concept or role names x and y:

$$x \sim_{\text{default}} y = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

   However, other primitive measures are imaginable and useful. For example, one might want to express that two amounts Medium and High are more similar than Low and High, which can be achieved by using a primitive measure with $\mathsf{Medium} \sim_p \mathsf{High} = 0.5$ and $\mathsf{Low} \sim_p \mathsf{High} = 0$.

2. A *weighting function* $g : N_C \cup N_R \to \mathbb{R}_{>0}$ to prioritize different features in the similarity measure. We give a default weighting function g_{default}, that assigns 1 to every concept and role name, but again, other weighting functions can be useful for certain cases.
3. A *discounting factor* w is a constant that allows for discounting of successors, and should have a value $0 < w < 1$.

Definition 3. *Given a primitive measure $\sim_p$, a weighting function g and the discounting factor w, the* interpretation similarity measure $\sim_i$ *is defined as:*

$$p \sim_i q = \frac{sim_{CN}(p,q) + sim_{CN}(q,p) + sim_{SC}(p,q) + sim_{SC}(q,p)}{\sum_{C \in CN(p)} g(C) + \sum_{D \in CN(q)} g(D) + \sum_{(r,p') \in SC(p)} g(r) + \sum_{(s,q') \in SC(q)} g(s)},$$

where

$$sim_{CN}(p,q) = \sum_{A \in CN(p)} \max_{B \in CN(q)} g(A)(A \sim_p B), \text{ and}$$

$$sim_{SC}(p,q) = \sum_{(r,p') \in SC(p)} \max_{(s,q') \in SC(q)} g(r)(r \sim_p s)\big(w + (1-w)(p' \sim_i q')\big).$$

If all of the sets $CN(p)$, $CN(q)$, $SC(p)$, and $SC(q)$ are empty for pointed interpretations p, q, we define $p \sim_i q = 1$.

Note that $\sim_i$ does not necessarily yield an equivalence closed or equivalence invariant CSM. To regain these properties, one can first normalize the interpretations before applying the $\sim_i$. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is in *interpretation normal form* if there are no elements $a, b, c \in \Delta^{\mathcal{I}}$, $b \neq c$, such that $\{(a,b),(a,c)\} \subseteq r^{\mathcal{I}}$ and for all concepts C with $b \in C^{\mathcal{I}}$ also $c \in C^{\mathcal{I}}$ holds; i.e., no node has two successors for the same role name whose instantiators are in a subset relation[3]. Any interpretation $\mathcal{I}$ can be transformed into normal form in polynomial time by removing all redundant successors.

Let $\mathcal{I}'_{C,\mathcal{T}}$ and $\mathcal{I}'_{D,\mathcal{T}}$ denote canonical models in interpretation normal form, then the CSM $\sim_c$ is defined as follows:

$$C \sim_c D = (\mathcal{I}'_{C,\mathcal{T}}, d_C) \sim_i (\mathcal{I}'_{D,\mathcal{T}}, d_D).$$

We showed that $\sim_c$ is indeed symmetric, equivalence invariant, and equivalence closed [9].

Example 1. Consider the concepts:
$C_{\mathsf{ex}} = \mathsf{Server} \sqcap \exists \mathsf{hasLoad}.\mathsf{Medium} \sqcap \exists \mathsf{provides}.(\mathsf{VideoStreamService} \sqcap \mathsf{Service})$ and
$D_{\mathsf{ex}} = \mathsf{Server} \sqcap \exists \mathsf{hasLoad}.\mathsf{Low} \sqcap \exists \mathsf{provides}.(\mathsf{DBService} \sqcap \mathsf{Service} \sqcap \exists \mathsf{queryLang}.\mathsf{SQL})$.
We use the primitive measure $\sim_p$, which is almost the default primitive measure, except for $\mathsf{Low} \sim_p \mathsf{Medium} = \mathsf{Medium} \sim_p \mathsf{High} = 0.5$ instead of 0. We also

[3] Formally, one describes this using the notion of a *simulation* between b and c in $\mathcal{I}$, see [13] for more details.

use the default weighting function g_{default} and the discounting factor $w = 0.8$. To compute the similarity between C_{ex} and D_{ex}, we have to compute their normalized canonical models (w.r.t. to the empty TBox). Based on these, we obtain: The $\mathsf{hasLoad}$-successors of C_{ex} and D_{ex} have a similarity of 0.5, since $\mathsf{Medium} \sim_p \mathsf{Low} = 0.5$. Both $\mathsf{provides}$-successors of C_{ex} and D_{ex}, are instances of $\mathsf{Service}$, while the concept names $\mathsf{VideoStreamService}$ and $\mathsf{DBService}$ have no correspondence. Similarly, only the $\mathsf{provides}$-successor of D_{ex} has an existential restriction, resulting in a value of 0 for sim_{SC} in both directions. Overall, this yields a similarity of $\frac{(1+0)+(1+0)+0+0}{2+2+0+1} = 0.4$ for the two services. Using this, we can finally compute the similarity between C_{ex} and D_{ex}:

$$\begin{aligned}
&\text{sim}_{\text{CN}}(C_{\mathsf{ex}}, D_{\mathsf{ex}}) = \text{sim}_{\text{CN}}(D_{\mathsf{ex}}, C_{\mathsf{ex}}) = 1\\
&\text{sim}_{\text{SC}}(C_{\mathsf{ex}}, D_{\mathsf{ex}}) = \text{sim}_{\text{SC}}(D_{\mathsf{ex}}, C_{\mathsf{ex}}) = (0.2 + 0.8 \cdot 0.5) + (0.2 + 0.8 \cdot 0.4) = 1.12\\
&C_{\mathsf{ex}} \sim_c D_{\mathsf{ex}} = \frac{1 + 1 + 1.12 + 1.12}{1 + 1 + 2 + 2} = 0.707.
\end{aligned}$$

The procedure to compute relaxed instances of a query concept Q w.r.t. a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a threshold t, and the CSM $\sim_c$ has the following steps [9]:

1. Compute the canonical models $\mathcal{I}_{Q,\mathcal{T}}$ and $\mathcal{I}_{\mathcal{K}}$ of the query concept Q and the ABox $\mathcal{A}$ w.r.t. the TBox $\mathcal{T}$.
2. Transform these models into $\mathcal{I}'_{Q,\mathcal{T}}$ and $\mathcal{I}'_{\mathcal{K}}$.
3. Define a *maximal interpretation similarity* $\sim_i^{\max}$ between the normalized canonical models. The measure $\sim_i^{\max}$ is behaves like $\sim_i$, but chooses a subset of the concept names and successors for elements in the canonical model $\mathcal{I}_{\mathcal{K}}$ in a way to maximize the similarity value.
4. For each individual a occurring in $\mathcal{K}$, check if the maximal interpretation similarity between the element d_Q in $\mathcal{I}'_{Q,\mathcal{T}}$ and the element d_a in $\mathcal{I}'_{\mathcal{K}}$ is larger than the given threshold t. If so, a is a relaxed instance and is returned.

Formally, $\sim_i^{\max}$ is defined as the unique solution of the following equation system:

$$p \sim_i^{\max} q = \max_{\substack{C_q \subseteq \text{CN}(q)\\ S_q \subseteq \text{SC}(q)}} \frac{\left(\begin{array}{r}\text{sim}_{\text{CN}}\big(\text{CN}(p), C_q\big) + \text{sim}_{\text{CN}}\big(C_q, \text{CN}(p)\big)\\ + \text{sim}_{\text{SC}}\big(\text{SC}(p), S_q\big) + \text{sim}_{\text{SC}}\big(S_q, \text{SC}(p)\big)\end{array}\right)}{\sum_{C \in \text{CN}(p)} g(C) + \sum_{D \in C_q} g(D) + \sum_{(r,p') \in \text{SC}(p)} g(r) + \sum_{(s,q') \in S_q} g(s)}, \quad (1)$$

where

$$\begin{aligned}
&\text{sim}_{\text{CN}}(C_1, C_2) = \sum_{A \in C_1} \max_{B \in C_2} g(A)(A \sim_p B), \text{ and}\\
&\text{sim}_{\text{SC}}(S_1, S_2) = \sum_{(r,p') \in S_1} \max_{(s,q') \in S_2} g(r)(r \sim_p s)(w + (1-w)(p' \sim_i^{\max} q')).
\end{aligned}$$

We showed that using the maximal interpretation similarity indeed solves the problem of answering relaxed instance queries correctly, see [9].

Theorem 1 ([9]). *Individual a is a relaxed instance of Q w.r.t. $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, t, and $\sim_c$ iff $(\mathcal{I}'_{Q,\mathcal{T}}, d_Q) \sim_i^{max} (\mathcal{I}'_{\mathcal{K}}, d_a) > t$.*

We showed an upper complexity bound of NP by (non-deterministically) translating the equation system that defines $(\mathcal{I}'_{Q,\mathcal{T}}, d_Q) \sim_i^{\max} (\mathcal{I}'_{\mathcal{K}}, d_a)$ into a linear programming problem of polynomial size and solving it. However, this approach is not practical. Instead, ELASTIQ implements an iterative approach, that refines the similarity values and converges to $\sim_i^{\max}$ in the limit. This approach which we present next is sound, as it converges from below, but not necessarily complete.

4 The ELASTIQ Reasoner

ELASTIQ is the first implementation for answering instance queries relaxed by a similarity measure. Given an $\mathcal{EL}$-ontology, an $\mathcal{EL}$-query concept, an instantiation of $\sim_c$ and a value for a threshold, ELASTIQ computes a result set of ABox individuals, where each of these individuals is relaxed instance. The CSM employed here is fixed to $\sim_c$ as defined in Section 3, but it and can be adjusted by a custom weighting function, primitive similarity measure and the discounting factor. Computing an answer to a relaxed instance query by ELASTIQ consists of four main steps.

Step 1: Global preprocessing. The canonical model $\mathcal{I}_{\mathcal{K}}$ of the ABox and the TBox is generated by the use of a standard DL reasoner, currently ELASTIQ uses the ELK system [11].

Step 2: Local preprocessing. The canonical model of the query concept w.r.t. the TBox $\mathcal{I}_{Q,\mathcal{T}}$ is generated—as in Step 1 by the use of ELK.

ELASTIQ distinguishes the two preprocessing steps for the sake of computing several relaxed instance queries against the same KB faster. Obviously, $\mathcal{I}_{\mathcal{K}}$ does not depend on the query concept and can therefore be reused for every subsequent queries. The model $\mathcal{I}_{Q,\mathcal{T}}$, however, needs to be recreated for every different query concept Q. In both steps we use the ELK reasoner [11] to compute classification and realization of the ontology, and then retrieve subsumption and instance relationships from the results. ELASTIQ only needs to consider those domain elements that are reachable from elements representing ABox individuals and thus can be used by the main algorithm. Similarly, for $\mathcal{I}_{Q,\mathcal{T}}$ ELASTIQ creates only those domain elements that are reachable through successor relations from d_Q. The normal forms of the canonical models are computed on demand. Finally, Step 2 also initializes the data structure for the main computation in Step 3.

Step 3: Computing the maximal interpretation similarity $\sim_i^{max}$. Recall, that ELASTIQ implements an iterative approach, that refines the similarity values and converges to $\sim_i^{\max}$ in the limit. Thus the main computation yields a sequence of matrices, each representing an iteration of the computation. The rows of such a matrix M_j represent domain elements from $\mathcal{I}_{Q,\mathcal{T}}$ and the columns domain elements from $\mathcal{I}_{\mathcal{K}}$. The values inside each cell of M_j, are identified by two domain elements $d \in \Delta^{\mathcal{I}_{Q,\mathcal{T}}}$ and $e \in \Delta^{\mathcal{I}_{\mathcal{K}}}$, and converge towards $(\mathcal{I}_{Q,\mathcal{T}}, d) \sim_i^{\max} (\mathcal{I}_{\mathcal{K}}, e)$

for $j \to \infty$ [9]. Instead of computing the similarity values for all pairs of elements from the canonical models in each iteration, ELASTIQ restricts the entries in M_j to those elements that are reachable from (elements representing) ABox individuals by paths in $\mathcal{I}_\mathcal{K}$. To this end, M_0 is initialized with one row (for d_Q) and as many columns as there are individuals in the ABox. The set of columns is extended with new elements (d', e') if there exists an element (d, e) in M_0 such that d and d' are connected in $\mathcal{I}_{Q,\mathcal{T}}$ via some role r, and e and e' are connected in $\mathcal{I}_\mathcal{K}$ via some role s. Since the canonical models $\mathcal{I}_{Q\mathcal{T}}$ and $\mathcal{I}_\mathcal{K}$ are finite, the size of M_0 is bounded by $|\Delta^{\mathcal{I}_{Q,\mathcal{T}}}| \cdot |\Delta^{\mathcal{I}_\mathcal{K}}|$. Once all reachable pairs have been added to M_0, it contains values exactly for those pairs that are necessary for computing similarities between the domain elements that we are interested in—namely similarities of the query concept and each ABox individual (d_Q, d_{a_i}).

Each iteration $j + 1$ creates a new matrix M_{j+1}, and computes the values by applying Equation (1) to the values in M_j. ELASTIQ needs only to keep the current matrix M_{j+1} and the last one M_j $(j \geq 0)$ in memory. The iterations for the refinement of similarity values proceeds until one of the following termination criteria is met:

- the maximal amount of iterations i_{max} specified by the user is reached; or
- no interesting similarity value (d_Q, d_a) has changed during the last iteration by more than a relative factor specified by the user.

Step 4: Comparison with t. After the iteration stopped, the interesting similarity values $M_j(d_Q, d_a)$ are compared to the input threshold t and the answer set of individuals is compiled. This set is then listed in descending order of similarity.

4.1 Optimizations for Computing Relaxed Instances

A naive implementation of the algorithm can hardly compute relaxed instances for reasonably large ontologies in acceptable time. As mentioned before, a highly effective optimization is the reuse of $\mathcal{I}_\mathcal{K}$ for multiple queries. Since ABoxes are usually much larger than query concepts, the model $\mathcal{I}_\mathcal{K}$ is also be much more costly to create than the models $\mathcal{I}_{Q,\mathcal{T}}$. Additionally, the normalization of canonical models can be done more efficiently than by computing simulations to determine unnecessary role-successors. Before adding a domain element d_C as an r-successor to some element d_D ELASTIQ checks whether there already exists an r-successor d_E for d_D such that $E \sqsubseteq C$. In this case normalization would eliminate d_C, thus avoiding the introduction of d_C (and its role successors) improves the runtime of canonical model generation further. Similarly, when adding d_C as an r-successor to d_D, ELASTIQ eliminates all r-successors d_E of d_D if $C \sqsubseteq E$.

During the generation of the canonical models ELASTIQ performs many subsumption checks. Although ELK is currently one of the fastest reasoners for $\mathcal{EL}$, caching of sub- and superclass relations yielded a great performance boost, since ELASTIQ needs to access sub- and superclass relationships for the same class several times.

The algorithm from Section 3 suggests to iterate over all subsets of CN and SC successors in order to find the maximal similarity. This exponential procedure

can be improved by looking at the primitive similarities between elements. Let $d \in \mathcal{I}_{Q,\mathcal{T}}$ and $e \in \mathcal{I}_{\mathcal{K}}$. By definition of $\sim_i^{\max}$ we are looking for those subsets of the concept names and successors of e that maximize the similarity. Instead of iterating over all subsets of $\mathrm{CN}(e)$ to find the best pairing, we showed that if $B \in \mathrm{CN}(e)$ such that $\exists A \in \mathrm{CN}(d)$ with $A \sim_p B = 1$, we can always keep B in the subset of $\mathrm{CN}(e)$, because it will always increase the similarity. Conversely, if $B' \in \mathrm{CN}(e)$ such that $\forall A \in \mathrm{CN}(d)$, then $A \sim_p B' = 0$, and B' can be left out of the subset of $\mathrm{CN}(e)$, since it cannot increase the similarity. Analogously, we can remove (s, q) from $\mathrm{SC}(e)$ if for all $(r, p) \in \mathrm{SC}(d)$ we have $r \sim_p s = 0$. This can dramatically reduce the number of subsets to be checked. In fact, for the default primitive measure, this means that the best subset of concept names can always be computed in linear time by checking each concept name in $\mathrm{CN}(e)$ separately.

4.2 Evaluation

Our preliminary performance evaluation of ELASTIQ used different versions of the GeneOntology that describe *schizosaccharomyces pombe*—some species of yeast. These ontologies ranged from 9,157 concept names and 34,875 individuals in the first version to 51,949 concept names and 289,206 individuals for the 15th version. The sizes of canonical models ranged from 77,941 to 602,548 elements.

We obtained our test ontologies by custom dataset generation provided by the Manchester OWL Corpus [14]. These GeneOntology versions are anonymised and therefore any contentual interpretation of our results is virtually impossible. We restricted our investigations solely to the performance of ELASTIQ and leave the intricate task of a quality assessment for future work. We discovered that for each individual e there exists a very fragmented concept assertion in the ABox of the form $\exists is\_a.C_e(e)$, where the qualification C_e is rarely larger than 3 conjuncts with a role-depth of at most 2.

Our test suite contains 10 randomly generated query concepts with increasingly complex structures. These queries were built over the common signature of versions 1–15 of the GeneOntology (approximately 1,000 concept names and 4 roles). The smallest query (Query 1) only contained 6 concept and role names and had a role-depth of 2, while the Query 10 had a size of 670 and a role-depth of 5. Due to the plain structure of concept assertions we wrapped each query concept Q_i with $\exists is\_a.Q_i$ in order to provoke a more complex computation. For these queries, the sizes of the canonical models $\mathcal{I}_{Q,\mathcal{T}}$ ranged from 2 to 236 elements. We evaluated the queries for the default primitive measure and weighting function, and counted the number of relaxed instances for a threshold of $t = 0.333$. The test system had a 1800 MHz dual core processor AMD Turion II Neo and 6 GB of RAM. Figure 1 shows the runtime of ELASTIQ for answering all 10 relaxed instance queries w.r.t. each ontology version. The high runtimes for ontology versions 11, 12, and especially 13–15 is mainly due to the increase of the size of the canonical model $\mathcal{I}_{\mathcal{K}}$. Most queries returned a lot more relaxed instances for the ontologies 11–15 than for ontologies 1–10. Queries 8 and 9 returned the largest number of relaxed instances, up to over 200,000 for Query 8 evaluated on nnotations15.owl.

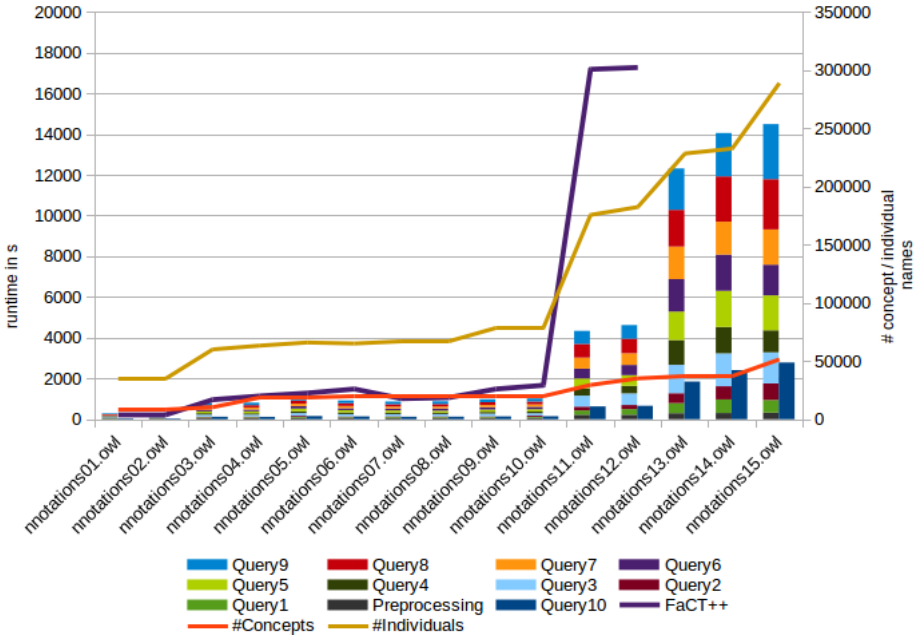

Fig. 1. ELASTIQ's runtime for answering relaxed instance queries in different versions of the GeneOntology

When breaking down the times for preprocessing and the query answering further, it shows that the preprocessing time is dominated by the flattening of the ontology, the reasoning done by ELK, and the construction of the canonical model $\mathcal{I}_{\mathcal{K}}$, while the time to construct the canonical models $\mathcal{I}_{Q,\mathcal{T}}$ is negligible. However, the overall query answering time is largely spend on Step 3, i.e., the iterations to compute the maximal similarity.

ELASTIQ performs ABox realization to obtain $\mathcal{I}_{\mathcal{K}}$ and in addition a kind of relaxed ABox realization for the query concepts in the test suite. Now, while it is clear that ELASTIQ is slower than ELK for ABox realization, it showed, suprisingly, that this does not need to be the case for other optimized DL reasoners. We compared ELASTIQ's overall reasoning times with the ABox realization times of the commonly used FaCT++ reasoner [18]. Figure 2 shows that ELASTIQ mostly performed better than FaCT++, although solving a more complex task.[4] However, computation times of more than a minute for 10 relaxed queries over ABoxes with 1,000 individuals still calls for further improvement.

5 Conclusions and Future Work

In this paper we investigated the novel inference of answering relaxed instance queries. These queries can be gradually relaxed by varying the threshold, while

[4] Note that FaCT++ classification resulted in an error for ontologies 13–15.

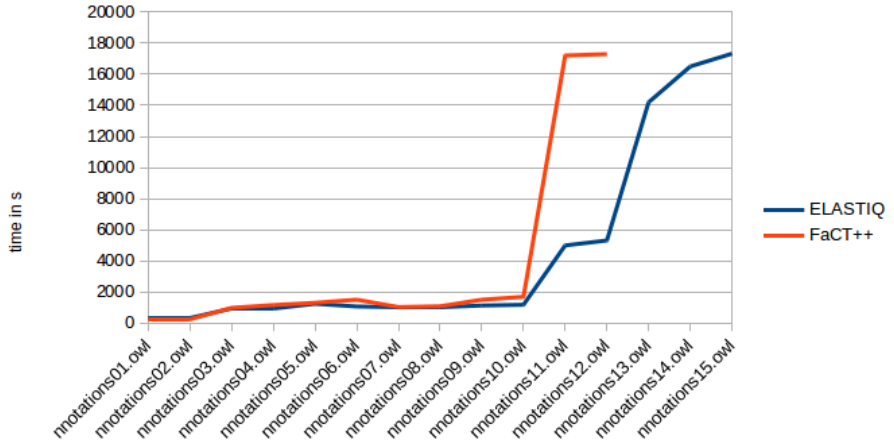

Fig. 2. Runtime of ELASTIQ compared to FaCT++ ABox Realization times.

the similarity measure allows to specify which parts of the query can be relaxed and which should be kept. We devised a concept similarity measure, $\sim_c$, that works for general $\mathcal{EL}$-TBoxes, and showed how to relax instance queries using this measure. We presented ELASTIQ, a prototype sytem for relaxed instance query answering, some straight-forward, but highly effective optimization techniques, and gave a first performance evaluation using different queries on increasingly complex versions of the GeneOntology.

It turned out that for ontologies with large ABoxes, ELASTIQ is still not fast enough. We want to explore further optimizations for the computation of $\sim_i^{max}$. Currently, the matrices converge to $\sim_i^{max}$ from below. With upper bounds on the maximal similarity, it would be possible to prune computation early on individuals that are certainly not relaxed instances. While currently the algorithm decides which individuals are certainly relaxed answers, an upper bound could also be used to determine which individuals are certainly not relaxed answers, therefore making the approach not just sound, but complete. We also need to perform further evaluations for the performance on ontologies and query concepts from other domains, but also to evaluate the quality of the answers returned by ELASTIQ in regard of the different CSMs in use.

Currently, one needs to specify a threshold, above which individuals are considered as relaxed answers. This threshold approach guarantees a minimal similarity and hence quality of the result, but it is hard to predict how many relaxed answers a query would return for a certain threshold. In cases where just the first few most similar relaxed answers are of interest, top-k answering would be more useful. We are investigating to efficiently implement this type of answering mechanism. Finally, it would be useful to not just consider instance queries, but more expressive query types as well. We are currently working on the theoretical basis for relaxing conjunctive queries.

References

1. A. Borgida, T. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In *Proc. of the 2005 Description Logic Workshop (DL 2005)*, volume 147 of *CEUR Workshop Proceedings*, 2005.
2. S. Borgwardt, F. Distel, and R. Peñaloza. How fuzzy is my fuzzy description logic? In B. Gramlich, D. Miller, and U. Sattler, editors, *Proceedings of the 6th International Joint Conference on Automated Reasoning (IJCAR'12)*, volume 7364 of *Lecture Notes In Artificial Intelligence*, pages 82–96. Springer-Verlag, 2012.
3. S. Borgwardt and R. Peñaloza. Undecidability of fuzzy description logics. In G. Brewka, T. Eiter, and S. A. McIlraith, editors, *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-12)*, pages 232–242. AAAI Press, 2012.
4. M. Cerami and U. Straccia. On the (un)decidability of fuzzy description logics under lukasiewicz t-norm. *Inf. Sci.*, 227:1–21, 2013.
5. C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. In *Proc. of Convegno Italiano di Logica Computazionale, CILC05*, 2005.
6. A. Ecke. Similarity-based relaxed instance queries in $\mathcal{EL}^{++}$. In *Proceedings of the First Workshop on Logics for Reasoning about Preferences, Uncertainty, and Vagueness*, CEUR-WS. CEUR, 2014.
7. A. Ecke, R. Peñaloza, and A.-Y. Turhan. Towards instance query answering for concepts relaxed by similarity measures. In *Workshop on Weighted Logics for AI (in conjunction with IJCAI'13)*, Beijing, China, 2013.
8. A. Ecke, R. Peñaloza, and A.-Y. Turhan. Answering instance queries relaxed by concept similarity. In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, pages 248–257. AAAI Press, 2014.
9. A. Ecke, R. Peñaloza, and A.-Y. Turhan. Similarity-based relaxed instance queries. *Journal of Applied Logic*, 2015. In press.
10. T. Gene Ontology Consortium. Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
11. Y. Kazakov, M. Krötzsch, and F. Simančík. The incredible ELK: from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *J. Autom. Reasoning*, 53(1):1–61, 2014.
12. K. Lehmann and A.-Y. Turhan. A framework for semantic-based similarity measures for $\mathcal{ELH}$-concepts. In L. F. del Cerro, A. Herzig, and J. Mengin, editors, *Proc. of the 13th European Conf. on Logics in A.I. (JELIA 2012)*, Lecture Notes In Artificial Intelligence, pages 307–319. Springer, 2012.
13. C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, 45(2):194–228, 2010.
14. N. Matentzoglu, S. Bail, and B. Parsia. A snapshot of the OWL web. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, pages 331–346, 2013.
15. B. Suntisrivaraporn. A similarity measure for the description logic $\mathcal{EL}$ with unfoldable terminologies. In *5th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pages 408–413, 2013.
16. S. Tongphu and B. Suntisrivaraporn. A non-standard instance checking for the description logic $\mathcal{ELH}$. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2014)*, Rome, Italy, 2014.

17. S. Tongphu and B. Suntisrivaraporn. On desirable properties of the structural subsumption-based similarity measure. In T. Supnithi, T. Yamaguchi, J. Z. Pan, V. Wuwongse, and M. Buranarach, editors, *Proceedings of the 4th Joint International Conference on Semantic Technology, (JIST 2014)*, volume 8943 of *LNCS*, pages 19–32. Springer, 2014.
18. D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In *Proceedings of the Third International Joint Conference on Automated Reasoning*, IJCAR'06, pages 292–297, Berlin, Heidelberg, 2006. Springer-Verlag.