Precision and Recall for Ontology Enrichment

Andreas Faatz and **Ralf Steinmetz** [1]

Abstract. Ontology enrichment algorithms propose new concepts to given concepts in a domain specific ontology. The paper is dedicated to the quality of ontology enrichment algorithms in terms of precision and recall. We remain open for the proposition of several new concepts to a given one instead of exactly one to a given one. Our first contribution is the generalization of known quality measures concerning recall for robust enrichment algorithms. In order to achieve independence from user evaluations and only rely on ontologies, text corpora and the output of the algorithms, there will be no appropriate explanation of precision in automatic evaluation - but hints on precision. Our second contribution is an explanation of these hints on precision in corpus-based ontology enrichment. Finally we show, which measures allow a straightforward computation.

1 Overview

Ontologies consist of well defined concepts from a knowledge domain. In an ontology, the concepts are interconnected by semantic relations. Ontologies provide semantics, which can be used by a broad range of applications such as for example search and retrieval, semantically enhanced web services and software agent communications.
There exist several accepted work-flows for ontology engineering, for example as collaborative processes of domain experts [HJ02]. All state-of-the-art definitions of ontology engineering work-flows essentially are open to so-called ontology enrichment techniques.
Ontology enrichment techniques are automatic processes, which generate extensions of the evolving ontology and propose these extensions to the ontology engineers. In general the extensions can include:

- new concepts to be integrated
- new relations to be instantiated between existing concepts
- corrections of existing concepts and relations formal explanations of how to map and merge several ontologies

Previous work was dedicated to ontology enrichment algorithms, which propose concepts according to linguistic regularities in domain text corpora [FS02], [FHS+02]. Along with the concepts, the algorithm suggests a place among the concepts in the ontology, where the semantics of the new concept might fit well.
This paper compiles the crucial points of measuring the quality of ontology enrichment for algorithms working with lexical items as propositions. Our aim is to classify ideas for a foundation of quality measures in analogy to precision and recall for text retrieval problems. A special issue is a type of a quality measurement, which works with realistic requirements in the sense of the ontology engineering process and are thus independent from user evaluations. Work on the quality measures is dedicated to two major benefits, namely:

- several algorithmic approaches become comparable
- fine tuning of ontology enrichment approaches itself becomes possible - analogous to other applications of machine learning like search engines or text categorization.

The most common definition of an ontology states that an ontology is a shared conceptualization of a (knowledge) domain [Gr93]. Although this definition concentrates on the notion of conceptualization, a more technical variant clearly describing the interconnections of the domain concepts will be needed. Therefore our paper is organized as follows. In section 2 we refer to a definition with minimal requirements on components of an ontology. In turn these minimal requirements will be applied in our precision and recall measures.
In section 3 we will introduce ontology enrichment.
Section 4 will develop recall measures for ontology enrichment and we will show approaches to precision aspects. We end with a brief conclusion and with future work.

2 Ontology definitions and assumptions

An ontology is a set of concepts ordered by a subconcept relation. Moreover there exists a set of relation names together with a restriction for each one of them: the restriction expresses, which subconcepts of which superconcept are allowed at the i-th place of a relational tupel. We refer to Stumme and Maedche [SM01] for such an ontology definition.

Definition 1 (Ontology) *A (core-) ontology is a 4-tupel* $\Omega := (B, \leq, R, \sigma)$*, where* B *and* R *are finite sets,* $\leq$ *a transitive, reflexive and asymmetric Relation on* $B \times B$ *(*$\leq \subseteq B \times B$*),* $\sigma : R \mapsto B^+$ *a mapping, which maps each* $r \in R$ *to a pair* $(B_1(r), B_2(r))$ *with* $B_1 \subseteq B$ *and* $B_2 \subseteq B$*. We call the* $b \in B$ *concepts,* $\leq$ *the subconcept relation and the* $r \in R$ *semantic relation names. Furthermore there exists an abstract root concept* $\top$ *for all concepts in* B*:* $\exists(\top \in B)\forall(b \in B) : \; b \leq \top$.

Definition 1 does not imply, that there is at least one natural language string naming every concept. Therefore we also add the following assumption to the definition of an ontology:

Assumption 1 (Concept descriptor) *Every concept* $b \in B$ *of the ontology from definition 1 has at least one natural language string as a name. We call this string, which can consist of one or several words, the descriptor* $d(b)$ *of the concept.*

With assumption 1, a huge class of ontology construction and enrichment problems will find a formal basis.

[1] Darmstadt University of Technology, Darmstadt, Germany email: Andreas.Faatz, Ralf.Steinmetz@kom.tu-darmstadt.de

3 Ontology enrichment

In contrast to conceptual clustering techniques, ontology enrichment works with given knowledge representations. It is meant to support real-world ontology engineering processes. That means that we introduce ontology enrichment as a group of approaches which

- generate for a given ontology Ω additional concepts as propositions,
- find a place for these propositions in Ω,
- operate with statistical data about the usage of the descriptors (names of concepts) of the ontology in a text corpus (in a collection of texts).

The background of this decision comes from the way an ontology is constructed by domain experts: instead of collecting and ordering lists of related terms and defining a concept by such a group, an ontology engineer might also think of a concept fully described by its name. In such a way we might end up with exactly one name for each concept.
In contrast to our assumption 1 existing approaches do not necessarily work with lexicalized concepts and lexical descriptors for propositions. For example

- [TRB02] applies description logic to an automatic instantiation of ontologies
- [GW02] introduces meta-properties and logic for ontology maintenance - with new concepts to be integrated
- [SM01] applies Formal Concept Analysis to define new concepts for ontology merging

However our formalization and the related approaches from word clustering follow assumption 1. This provides a basis for precision and recall definitions without the need of user evaluations.

3.1 Formalization of ontology enrichment

Before we start our work on formalizing the notion of enrichment quality by defining numeric measures, the corpus-based ontology enrichment problem itself must be explained in a formal way.
Formally we define:

Definition 2 (Ontology enrichment) *Let ξ be a text corpus, that means, a collection of written or spoken text documents, which are processable for natural language analysis. Let $B(\xi)$ a set of words and phrases from ξ. An ontology enrichment algorithm, is an algorithm which takes a given ξ and a given ontology Ω as input and produces for each $b \in B$ a set $P(b) \subseteq B(\xi)$ as output. We call $P(b)$ the set of propositions for b.*

The sets of propositions for an ontology enrichment algorithm can be derived by similarity functions [FS02], [FHS+02]. For given Ω and ξ let $s_0 : B \times B \rightarrow R_0^+$ be a similarity function mapping each pair of concepts from the ontology to a similarity value. The algorithm in [FHS+02] extends s_0 to a general similarity function

$$s : B(\xi) \times B(\xi) \rightarrow R_0^+ \tag{1}$$

and the sets of propositions P(b) for a given $b \in B$ and a real threshold T can thus be defined as

$$P(b) = P(b, T) = \{b_i \in B(\xi) | s(b, b_i) \geq T\} \tag{2}$$

We see from equations (1) and (2), that the main parameters for the quality of an enrichment algorithm in the sense of [FHS+02] stem from the definition of the similarity functions and the variation of the thresholds. If we define quality measures related to precision and recall we would expect a higher recall for lower T, because

$$|P(b, T_1)| \geq |P(b, T_2)| \Leftrightarrow T_1 \geq T_2 \tag{3}$$

T must not be expressed as one absolute numeric value for all ontology enrichment problems, but as a variable depending on the structure of the ontology and on the chosen concept b.
The core ideas of ontology enrichment differ from automatic ontology construction, which can be supervised or unsupervised (e.g. [N99]).
Let us visualize the different paradigms of approaches related to clustering and an ontology enrichment approach. From Definition

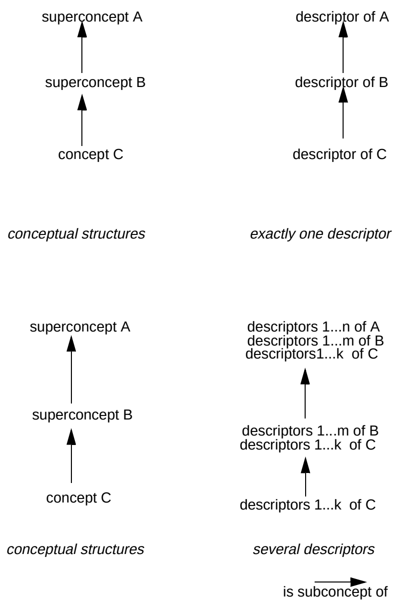

Figure 1. Clustering and enrichment: the initial structures

1 we adopt the fact, that it consists of concepts, which are ordered by the relation $\leq$ as superconcepts and subconcepts. The below figure shows a dummy branch from the type of ontologies we enrich on the left. We find the following situation. For each concept A, B and C there exist several descriptors, in a set-theoretic notation we find the sets of descriptors $D(A) := \{d_1(A), ..., d_n(A)\}$ for A, $D(B) := \{d_1(B), ..., d_m(B)\} \cup D(A)$ for B, and $D(C) := \{d_1(C), ..., d_k(C)\} \cup D(B)$ for C and therefore

$$D(A) \subseteq D(B) \subseteq D(C) \tag{4}$$

A change of the lower part Figure 1 (on the right), such that the sets of descriptors do not form chains like the one in equation (4) would yield the typical enrichment perspective from [MPS02] or [AL01]. These approaches work with structures to be found in the synset hierarchies of thesauri as for example WordNet [WordNet]. For each concept in WordNet, there exists set of synonyms, which does not appear in the superconcepts.

All approaches with several descriptors underly Assumption 1, but the assumption of descriptor lists turns out be a strong one, if we take the ontology engineering process into account. Especially non experts in formalization and conceptualization will establish hierarchies of single descriptors from scratch - without the additional effort of grouping similar descriptors and placing them into a hierarchy of such groups. Furthermore we will see, that the existence of several descriptors has an impact on evaluation techniques.

3.2 Ontology enrichment example

Let us finally show two brief and simplified examples of an ontology enrichment technique, which operate with collocation data of the text corpus. Let Ω_{ex} be given with $B_{ex} := \{\top, disease, diarrhoea\}$ and

$$diarrhoea \leq disease \leq \top \tag{5}$$

Furthermore, consider the following example collocations from a text corpus ξ_{ex}.

disease:heart (1543), AIDS (1063), patients (962), Alzheimer's (773), cancer (544), coronary (429), blood (354), Parkinson's (311), symptoms (304), virus (300)
diarrhoea: symptoms (30), virus (11) vomiting (5) nausea(3) salmonella(1)

The example should be read as "heart" co-occurred with "disease" in ξ_{ex} 1543 times, "symptoms" co-occurred with "diarrhoea" in ξ_{ex} 30 times et cetera. In this case, co-occurrence refers to being words in the same sentence.

An ontology enrichment approach in the sense of [FHS+02] would identify the set {"symptoms", "virus"} as the co-occurring words of the concept descriptors for both *disease* and *diarrhoea*. The algorithm declares the similarity function s by the degree of similarity to a transformation of the vectors $(30, 11)$ (for *diarrhoea*) and $(304, 300)$ (for *disease*). In other words, a word with many co-occurrences of "symptoms" and "virus" is a potential proposition.

An ontology enrichment in the sense of [AL01] or [Y92] would identify set S of the co-occurring terms different from the ones in $\{"symptoms", "virus"\}$ and determine a significance interval $[a; b]$. If the number of co-occurrences is inside $[a; b]$, then $s \in S$ becomes a proposition.

4 Evaluation alternatives

This section consists of three parts. The first part motivates the need for automatic evaluation measures of ontology enrichment algorithms. The second part consists of related work - that means, it sketches strengths and weaknesses of existing automatized evaluation measures. The remainder of the section is dedicated to precision and recall measures with a special focus on the additional assumptions on the enrichment algorithm if we undertake automated evaluations.

4.1 Ontology enrichment processes and parameters

Holsapple and Joshi define an ontology engineering process [HJ02], where the ontology engineers systematically judge about the other ones' conceptual modelling. Each concept which is added to an existing ontology, is presented to at least another domain expert, who applies a number from a Likert scale to express the quality of the modelling. In principle, with such an approach one could also judge about ontology enrichment algorithms, but there exists a hurdle: before an ontology enrichment algorithm is applied to support a human-centric ontology engineering process, we need a fine-tuning of the algorithm concerning many open parameters. We now give an overview of these parameters.

First, there are different ways to construct the underlying domain specific text corpus for ontology enrichment. Especially in situations like the one described in [FS02] we meet the question, how World Wide Web (WWW) search results and textual archives should be transformed into a text corpus ξ to derive language regularities from. Generally spoken, corpus construction parameters become complicated with an increasing number of corpus sources. Moreover additional input parameters for ontology enrichment depend on the choice of the initial similarity function s_0. The ontology structure can be transformed by path-oriented (that means: paths in a graph with edges from B and relations from R for a given ontology $\Omega := \{B, \leq, R, \sigma\}$) measures like the ones from [Li03] or [FHS+02]. Another possibility are measures considering information content like the ones in [R99]. At this point we comment, that also the variation of the thresholds T can be highly dependent on the choice of s_0.

Unfortunately, if we extend s_0 to s further parameters occur. These depend on the similarity metaphor for the representation of word contexts in the sense of [L99]. Besides the similarity metaphor the question arises, if we should represent syntactic or window-based features from the word contexts [G92].

Finally with a growing ontology during an ontology construction process we face a growing need for decomposition techniques to keep enrichment algorithms performant. By a disjoint decomposition of Ω into $\Omega_1 := \{B_1, \leq, R, \sigma\}$,...,$\Omega_n := \{B_n, \leq, R_n, \sigma\}$ we mean a disjoint decomposition

$$B = \bigcup_{1 \leq i \leq n} B_i \tag{6}$$

together with at the corresponding restrictions of the relations from R_n and $\leq$.

From the vast amount of parameters influencing ontology enrichment we conclude, that ontology enrichment algorithm user evaluations are completely impossible - the bare amount of evaluation tasks is not acceptable in rapid ontology engineering. Moreover from our point of view there might occur another fundamental problem: the result of an enrichment algorithm might distract the evaluating user from the direction the ontology actually should have evolved. The other way around the evaluator might refuse helpful hints from the algorithmic output - both faults are generally known in information retrieval [BR99]. In ontology enrichment the problem becomes even worse as it is impossible to let a bigger group of users evaluate the results: by a central design requirement, domain ontologies are carved for a particular task and a particular application. Thus, an evaluation must always be aware of the task - thus, actually the ontology cannot be evaluated by a third party different from the ontology engineers. From both circumstances we conclude:

- only automatic evaluations of ontology enrichment meet the requirements of algorithmic tuning
- the automatization has to be aware of the task specific semantic direction, to which an ontology should evolve

4.2 Related automatic evaluation approaches

For our quality measures we refer to an approach using similarity thresholds instead of conceptual clustering techniques. The background of this decision comes from the way an ontology is constructed by domain experts: instead of collecting and ordering lists of related terms and defining a concept by such a group, an ontology engineer might also think of a concept fully described by its name. In such a way we might end up with exactly one name for each concept. This paradigm contrasts to the automated evaluations of [MPS02], where successively exactly one descriptor from a structure like the one in Figure 1 below is deleted and a test determines, if an ontology enrichment algorithm would propose it at this place again. If we delete a descriptor from a structure like the one in the upper part of figure 1, this technique fails, because we have changed the ontology (whereas in [MPS02] we do not meet the problem, that with a disappearing descriptor the whole conceptual information would disappear). Thus there is no direct application of [MPS02] to our evaluation setting.
Another evaluation paradigm is the cotopy measure [MS02]. Cotopy judges formally about the similarity between two ontologies. Also this measure is not directly applicable to our problem, because ontology enrichment does not establish an ontology, but an enrichment with propositions. The relations of the propositions are identified by a human. Thus with cotopy we would have to translate propositions into ontological relations. This would cause critical additional assumptions. Moreover cotopy judges about an ontology in total. An improved evaluation measure should be able to judge about both local and global phenomenons.
We now start constructing alternatives to the existing approaches.

4.3 Aspects related to precision

For an ontology enrichment algorithm we explicitly define the set of words or phrases, for which the similarity computations are calculated at all. With the notations from definition 2 we obtain

Definition 3 (candidate) *A candidate is a word or phrase $x \in B(\xi)$. A candidate detection mechanism defines the set $B(\xi)$*

To evaluate precision at all, we have to assume a candidate detection mechanism. Note again the difference between candidates and propositions: a candidate is a term, for which a co-occurrence vector is established and a similarity to concepts from the ontology is computed at all. A proposition is a candidate, which under the similarity definition is similar enough to be placed to a given concept. For the following facets of precision let a set $B(\xi)$ of candidates be given.
Regarding precision our central hypothesis is the following one: it is not possible to fully compute precision, if we do not know the semantics of each candidate. From the moment we consider words or phrases, which do not come from the given ontology, there is no automatic way of judging about their quality: from our point of view quality statements are only allowed for the descriptors we already met in the given concepts. Our attempt in the remaining sections is the following one: we define evaluation measures from observations, how an ontology enrichment algorithms causes an evolution of the ontology. If this evolution is comparable to the construction, which would be the outcome of human ontology engineering, we obtain a high quality of enrichment.

4.3.1 General likelihood of enrichment

Even if a high amount of recall can be gained by an ontology enrichment algorithm, the results might disappear in an overload of propositions. We conclude that for each family of parametrizations (see section 4.1) of an ontology enrichment algorithm we have to compare the number of propositions for the ontology. Technically spoken, we determine

$$|\{x \mid s(x,b) > T)|x \in B(\xi), b \in B\}| \tag{7}$$

. If this number is high, a lower precision *might* be given. Besides precision, the ontology engineer might wish to avoid an exaggerated number of propositions.

4.3.2 Percentage of true candidates and precision

At a first glance this is more an evaluation measure for the mechanism detecting candidates. This measure computes the ratio of terms in $B(\xi)$, which are (up to stemming) identic to the concept descriptors from the given ontology. If the ratio is high we might obtain a hint on a high quality of the way we defined the set of candidates. High quality in this sense rises the opportunities of the actual enrichment to work well.
Note that this evaluation measure contrasts to nearly all directions of conceptual natural language processing (e.g. [G92], [N99], [Y92]), which try to produce stemming for words to obtain a better statistics on their usage. An example of how different collocation information can turn out for singular and plural forms in the same corpus is listed below. The result was produced by querying the German Wortschatz project [wort], [Q98]. We list the first ten significant collocations together with their total.

disease:heart (1543), AIDS (1063), patients (962), Alzheimer's (773), cancer (544), coronary (429), blood (354), Parkinson's (311), symptoms (304), virus (300)

diseases: AIDS (279), cancer (265), infectious (199), sexually (177), transmitted (163), patients (154), lung (149), immune (140), chronic (113), genetic (112)
We observe remarkable collocation differences for both approaches. We conclude that we are able to produce true candidates via the lexicalization of the existing concept descriptors, but the true candidates may behave differently than the original descriptor in the corpus.
Unfortunately there are still other candidates left, which (to one extreme) could all over be equally good candidates or (to the other extreme) all be irrelevant. This is the reason why we - if we compare two versions of an ontology enrichment algorithm or two different ontology enrichment algorithms - should refer to the same candidates as long as we enrich the same ontology. From the point of view that is developed in [FS02] and [FHS+02] this is not self-evident, because the candidate detection mechanism can be a consequence of the feature selection. In terms of the example from section for two different settings of the algorithm (i.e. two different similarity metaphors) the enrichment - and in consequence a candidate detection - may purely depend on $\{"symptoms"\}$ in one case and purely on $\{"virus"\}$ in the other case. Instead, for evaluation purposes we should refer to candidate detection by all terms in $\{"virus","symptoms"\}$.

The second point is that we are forced to a pessimistic judgement of the other candidates. Thus we refer for the remainder of this section to a $B'(\xi)$, which consists of a $B(\xi)$, which we judge pessimistically and lexical variants $v(\Omega) := \{v_i(b)|b \in B, 0 \leq i \leq N,\}$ (i is the i-th variant and N the total of variants we permit) which we judge as good candidates:

$$B'(\xi) := B(\xi) \cup v(\Omega) \tag{8}$$

A precision evaluation method could work in the following way. Fix for each $b \in B$ a threshold $T_{v(b)}$ in such a way, that at least one $v_i(b)$ becomes a proposition for b and determine with the notation from equation (2)

$$\frac{\{x \in P(b, T_{v(b)})|x \in v_i(b), 0 \leq i \leq N\}}{|P(b, T_{v(b)}|} \tag{9}$$

This formula may be applied locally, that means for one b, or it may be used to average over the whole ontology. If for each $b \in B$ two lexical variants exist, an analogous evaluation method may start with thresholds, which produce at least two propositions for each concept.

4.3.3 Discrimination

This measure is another indicator of how well an algorithm separates between the candidates. If there are many candidates, but only a few propositions, this *might* be an indicator of precision. If

$$\frac{|\{x \mid s((r((x), r(c))) > T)\}|}{|B(\xi)|}$$

is low, then only relatively few of the candidates are actually selected as propositions. On the other hand this can again correspond to a bad candidate detection mechanism. A better solution for the determination of algorithms, which are *not* overly sensitive concerning the thresholds, is the maximization of the distance (i.e. minimization of the similarity) between the candidates, which are included as propositions and the candidates, which do not become propositions. In both enrichment examples from section 3.2. this is possible, because the similarity measures are an output of the enrichment algorithms. An example for a discrimination measure is

$$\sum_{b \in B} \min_{x \in B(\xi), x < T(b)} s(x, b) \tag{10}$$

For small values of this measure we obtain a clear discrimination in the sense, that the propositions are relatively robust concerning small variations of the thresholds $T(b)$. Discrimination is a second order goal and inspired by classification approaches like support vector machines [CVB+02]. In our sense discrimination should only be considered with a given bandwidth of precision and recall.

A common additional assumption of the precision definitions is the introduction of concrete candidate detection mechanism (defining $B(\xi)$ and even more its extension to a $B'(\xi)$ with lexical variants. The next section will describe other solutions, which are independent from such an assumption.

4.4 Aspects related to recall

For a given $\Omega := \{B, \leq, R, \sigma\}$ we generalize the idea of [SM01], but we also extinct existing concepts from a given ontology and collect them in a set C which becomes the candidate set and is independent from additional descriptors from the corpus. The idea of random choice of C can only be persuaded, if

$$\Omega' := \{B \setminus C, \leq, R', \sigma'\}, \tag{11}$$

where R' and σ' are restrictions of the relations and arities to remaining concepts in $B \setminus C$ is again an ontology. This holds because by transitivity of $\leq$ we may keep for instance a relation $k \leq m$, if $k \leq l \leq m$ was part of the original ontology. The restriction relation of the relation $\leq$ still remains transitive. Consequently we only have to claim $\top \notin C$.
Furthermore denote for $b_1, b_2 \in B$ by $d_\Omega(b_1, b_2)$ the shortest relational path along $\leq$ and its inversion $\geq$. Then we define our measure called 1-edge-recall for a given $c \in B$ as the ratio

$$\frac{|P(c) \cap \{b|d_\Omega(b, c) = 1\}|}{|\{b|d_\Omega(b, c) = 1\}|} \tag{12}$$

and more general n-edge-recall for a given $c \in B$ as the ratio

$$\frac{|P(c) \cap \{b|d_\Omega(b, c) \leq n\}|}{|\{b|d_\Omega(b, c) \leq n\}|} \tag{13}$$

Note that the n-edge-recall for $n > 1$ is not necessarily greater or equal than the n-edge-recall, as the denominator $\{b|d_\Omega(b, c) = n\}$ may grow faster than the corresponding enumerator.

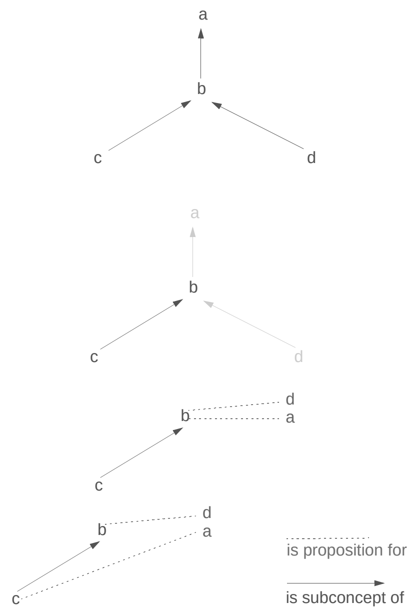

Figure 2. Recall example

Figure shows a recall example. From the above ontology two concepts d and a are removed and thus become candidates. We indicate this in the second depicted structure by brightening all subconcept relationships where d and a are involved. The remaining two structures represent enrichment outcomes. Propositions are depicted by drawing a dotted line between a concept and a candidate. In the first case of the two ontology enrichment procedures we obtain a 1-edge recall and a 2-edge-recall of 100 per cent each, in the second case

below we obtain 50 per cent for the 1-edge-recall and 100 per cent for the 2-edge recall.
By a variation on $|C|$ we can observe, if the enrichment algorithm is able to propose new concepts for more (greater $|C|$) or less (smaller $|C|$) complete ontologies.
Although the measures we presented in equations (12) and (13) work under the assumption that the given Ω expressed correct semantics, this assumption still seems a weaker one than a candidate detection mechanism and a lexical variation in our section on precision. This motivates the turn of (13) to a enrichment quality measure, which computes the ratio of proposition failures with the aim of measuring recall. If

$$\frac{|P(c) \cap \{b | d_\Omega(b, c) > n\}|}{|\{b | d_\Omega(b, c) > n\}|} \tag{14}$$

is high, we obtain many propositions, which are actually out of scope and our precision decreases. Thus, (14) can be understood as alternative to the precision measures from the subsection before. Finally, we mention that both n-edge recall and the proposition failure measure (14) may be averaged over the concepts of an ontology. Moreover, for n-edge-recall this average can be weighted with respect to the number of high influences from those concepts, which expect many propositions. Analogously, such a weighting scheme may be applied to the averaged proposition failure measure. For average computations, the threshold T can differ for each concept depending on how one or n edges of a graph (here B as nodes, $\leq$ and $>$ as edges).

5 Conclusion and outlook

We presented precision and recall measures as automatic approaches for ontology enrichment evaluations. The recall measures exploit the given ontology structures, are independent from additional assumptions on candidates and can be extended to a proposition failure measure. The latter one reflects precision. We conclude, that the strategy of a random extinct of concepts from an ontology to evaluate enrichment results is easier in the sense of applications. Nevertheless, empirical future work should examine the correlation between the recall measures and the precision we presented.
Last but not least, we need a better understanding of the consequences from different values of (12) through (14) for varying c. Such intra-ontological variations potentially yield new ontology engineering strategies.

REFERENCES

[AL01] Eneko Aguirre, Mikel Lersundi: Extraccin de relaciones lxicosemnticas a partir de palabras derivadas usando patrones de definicin. Procesamiento del Language Natural 27, 2001

[BR99] R. Baeza-Yates and B. Ribeiro-Neto: Modern Information Retrieval Addison Wesley, 1999.

[BNC00] G. Bisson, C. Nedellec and L. Caamero , Designing clustering methods for ontology building - The Mo'K workbench, Proceedings of the ECAI-2000 workshop on Ontology Learning, Berlin, Germany, 2000

[CVB+02] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, Sayan Mukherjee: Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46(1-3): 131-159 (2002)

[EN94] Ramez ElMasri, Shamkant B. Navathe: Fundamentals of Database Systems, 2nd Edition. Benjamin/Cummings 1994

[FS02] Andreas Faatz, Ralf Steinmetz: Ontology Enrichment with Texts from the WWW, Proceedings of the First International Workshop on Semantic Web Mining, European Conference on Machine Learning 2002, Helsinki 2002

[FHS+02] Andreas Faatz, Stefan Hoermann, Cornelia Seeberg, and Ralf Steinmetz. Conceptual Enrichment of Ontologies by means of a generic and configurable approach. In Proceedings of the ESSLLI 2001 Workshop on Semantic Knowledge Acquisition and Categorisation, August 2001

[GGW01] Ganesan, Prasanna; Garcia-Molina, Hector; Widom, Jennifer. Exploiting Hierarchical Domain Structure to Compute Similarity, Extended Technical Report, 2001

[G92] G. Grefenstette, Use of Syntactic Context to Produce Term Association Lists for Text Retrieval, SIGIR 1992, Copenhagen, Denmark, 1992

[Gr93] Thomas Gruber: A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220, 1993

[GW02] Nicola Guarino, Christopher A. Welty: Evaluating ontological decisions with OntoClean, Communications of the ACM 45 (2), 2002

[HJ02] Clyde W. Holsapple, K. D. Joshi: A collaborative approach to ontology design. Commun. ACM **45**(2): 42-47 ,2002

[H98] I. Horrocks , Using an expressive description logic: FaCT or fiction? Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98), San Francisco, California, USA, 1998

[L99] L. Lee ,Measures of Distributional Similarity , 37th Annual Meeting of the Association for Computational Linguistics, College Park, Maryland, USA, 1999

[Li03] Y. Li, Z.A. Bandar and D. McLean , An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources, IEEE Transactions on Knowlege and Data Engineering **15**, 2003

[MS02] Alexander Maedche and SteffenStaab, Measuring Similarity between Ontologies, Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, 2002

[MPS02] Alexander Maedche, Viktor Pekar, Steffen Staab: Ontology Learning Part One - On Discoverying Taxonomic Relations from the Web, in Ning Zhong et al. (eds) Web Intelligence. Springer, 2002

[N99] Claire Nedellec: Corpus-Based Learning of Semantic Relations by the ILP System Asium, Learning Language in Logic, Springer, 1999

[Q98] Quasthoff, Uwe: Tools for Automatic Lexicon Maintenance: Acquisition, Error Correction, and the Generation of Missing Values. In: Proceedings of the first International Conference on Language Resources and Evaluation, ELRA 1998

[R99] Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, Journal of Artificial Intelligence Research, 11, 1999

[SG83] G. Salton, M.J. McGill: Introduction To Modern Information Retrieval, McGraw Hill, New York, 1983

[SM01] Gerd Stumme, Alexander Mdche: FCA-MERGE: Bottom-Up Merging of Ontologies, Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001

[TRB02] Amalia Todirascu, Laurent Romary, Delila Bekhouche: Vulcain - An Ontology-Based Information Extraction System, Proceedings of: Natural Language Processing and Information Systems, 6th International Conference on Applications of Natural Language to Information Systems, NLDB 2002, Stockholm, Sweden, June 27-28, 2002

[WordNet] WordNet, a lexical database for the English language, *http://www.cogsci.princeton.edu/ wn/*

[wort] The German Wortschatz Project, *www.wortschatz.de*

[Y92] David Yarowsky: Word-Sense Disambiguation using Statistical Models of Roget's Categories Trained on Large Corpora, Proceedings of COLING-92, Nantes, France,1992