# Using Semantic Distances for Reasoning with Inconsistent Ontologies

Zhisheng Huang and Frank van Harmelen
{huang,Frank.van.Harmelen}@cs.vu.nl

Computer Science Department, Vrije Universiteit Amsterdam, The Netherlands

**Abstract.** Re-using and combining multiple ontologies on the Web is bound to lead to inconsistencies between the combined vocabularies. Even many of the ontologies that are in use today turn out to be inconsistent once some of their implicit knowledge is made explicit. However, robust and efficient methods to deal with inconsistencies are lacking from current Semantic Web reasoning systems, which are typically based on classical logic. In earlier papers, we have proposed the use of *syntactic relevance functions* as a method for reasoning with inconsistent ontologies. In this paper, we extend that work to the use of semantic distances. We show how Google distances can be used to develop *semantic relevance functions* to reason with inconsistent ontologies. In essence we are using the implicit knowledge hidden in the Web for explicit reasoning purposes. We have implemented this approach as part of the PION reasoning system. We report on experiments with several realistic ontologies. The test results show that a mixed syntactic/semantic approach can significantly improve reasoning performance over the purely syntactic approach. Furthermore, our methods allow to trade-off computational cost for inferential completeness. Our experiment shows that we only have to give up a little quality to obtain a high performance gain.

There is nothing constant in this world but inconsistency.
*-Jonathan Swift (1667-1745)*

## 1 Introduction

A key ingredient of the Semantic Web vision is avoiding to impose a single ontology. Hence, merging ontologies is a key step. Earlier experiments (e.g. [10]) have shown that merging multiple ontologies can quickly lead to inconsistencies. Other studies have shown how migration [18] and evolution [9] also lead to inconsistencies. This suggests the importance and omnipresence of inconsistencies in ontologies in a truly web-based world.

At first sight, it might seem that many ontologies are semantically so lightweight (e.g. expressible in RDF Schema only[6]) that the inconsistency problem doesn't arise, since RDF Schema is too weak to even express an inconsistency[1]. However, [17] has shown that on a closer look, many of these semantically lightweight

[1] besides the rather limited case of disjoint datatypes.

ontologies make implicit assumptions such as the Unique Name Assumption, or assuming that sibling classes are disjoint. Such implicit assumptions, although not stated, are in fact used in the applications that deploy these ontologies. Not making these disjointness assumptions explicit harms the re-usability of these ontologies. However, if such assumptions are made explicit, many ontologies turn out to be in fact inconsistent.

One way to deal with inconsistencies is to first diagnose and then repair them. [18] proposes a nonstandard reasoning service for debugging inconsistent terminologies. This is a possible approach, if we are dealing with one ontology and we would like to improve this ontology. Another approach to deal with inconsistent ontologies is to simply avoid the inconsistency and to apply a non-standard reasoning method to obtain answers that are still meaningful, even though they have been obtained from an inconsistent ontology. The first approach could be dubbed "removing inconsistencies", while the second could be called "living with inconsistencies". This latter approach is more suitable for an open Web setting. where one would be importing ontologies from other sources, making it impossible to repair them, and where the scale of the combined ontologies would be too large to make repair effective. Therefore, this paper investigates the latter approach, namely, the approach of reasoning with inconsistent ontologies.

The classical entailment in logics is *explosive*: any formula is a logical consequence of a contradiction. Therefore, conclusions drawn from an inconsistent knowledge base by classical inference may be completely meaningless. The general task of a system of reasoning with inconsistent ontologies is: given an inconsistent ontology, return *meaningful* answers to queries. In [12] we developed a general framework for reasoning with inconsistent ontologies, in which an answer is "meaningful" if it is supported by a selected consistent sub-ontology of the inconsistent ontology, while its negation is not supported. In that work, we used relevance based selection functions to obtain meaningful answers. The main idea of the framework is: (1) a relevance function is used to select some consistent sub-theory from an inconsistent ontology; (2) then we apply standard reasoning on the selected sub-theory to try and find meaningful answers; (3) if a satisfying answer cannot be found, the relevance degree of the selection function is made less restrictive, thereby extending the consistent sub-theory for further reasoning. In this way the system searches for increasingly large sub-theories of an inconsistent ontology *until the selected sub-theory is large enough to provide an answer, but not yet so large so as to become itself inconsistent.*

In [13, 11], several syntactic relevance based selection functions were developed. However, these approaches suffer several limitations and disadvantages. As we will show with a simple example later in this paper, such syntactic relevance functions are very sensitive to the accidental syntactic form of an ontology, which can easily lead to undesired conclusions on one syntactic form. A simple semantics preserving syntactic reformulation would have lead to the appropriate conclusion, but such careful design is unrealistic to require from knowledge engineers.

In this paper, we investigate the approach of *semantic relevance* selection functions as an improvement over the syntactic relevance based approach. We will examine the use of co-occurrence in web-pages, provided by a search engine like Google, as a measure of semantic relevance, assuming that when two concepts appear more frequently in the same web page, they are semantically more relevant. We will show that under this intuitive assumption, information provided by a search engine can be used for semantic relevance based selection functions for reasoning with inconsistent ontologies.

The main contributions of this paper are (1) to define some general formal properties of semantic relevance selection functions, (2) to propose the Google Distance as a particular semantic relevance function, (3) to provide an implementation of semantic relevance functions for reasoning with inconsistent ontologies in the PION system, (4) to run experiments with PION to investigate the quality of the obtained results, and (5) to highlight the cost/performance trade-off that can be obtained using our approach.

This paper is organised as follows. Section 2 briefly summarises the framework for reasoning with inconsistent ontologies. Section 3 introduces the notion of semantic relevance functions. Setion 4 presents a mixed approach which combine the advantages of both the syntactic approach and the semantic approach. Section 5 reports on our experiments of running PION on a realistic ontology before concluding the paper.

## 2 Reasoning with Inconsistent Ontologies

### 2.1 General Framework

Selection functions are central to the framework of reasoning with inconsistent ontologies. Such a selection function is used to determine which consistent subsets of an inconsistent ontology should be considered during its reasoning process. The selection function can either be based on a syntactic approach, like syntactic relevance [3], or based on semantic relevance. Examples of such semantic relevance are for example Wordnet distance [2], or (as we will propose in this paper) based on the co-occurrence of concepts in search engines like Google.

Given an ontology (i.e., a formula set) $\Sigma$ and a query $\phi$, a selection function $s$ returns at each step $k > 0$ a subset of $\Sigma$. Let $\mathbf{L}$ be the ontology language, which is denoted as a formula set. A selection function $s$ is then a mapping $s : \mathcal{P}(\mathbf{L}) \times \mathbf{L} \times N \to \mathcal{P}(\mathbf{L})$ such that $s(\Sigma, \phi, k) \subseteq \Sigma$.

In the following, we use $\Sigma \models \phi$ to denote that $\phi$ is a consequence of $\Sigma$ in the standard reasoning, and we will use $\Sigma \mathrel{|\!\approx} \phi$ to denote that $\phi$ is a consequence of $\Sigma$ in the non-standard reasoning. The values of non-standard inference are defined as {*Accepted, Rejected, Overdetermined, Undetermined*}, following the 4-valued schema by [1].

Figure 1 shows a strategy to compute $\Sigma \mathrel{|\!\approx} \phi$. This procedure is called a *linear extension* strategy because only one candidate $\Sigma''$ is chosen, and alternatives are not considered. This is attractive because the reasoner doesn't need to keep

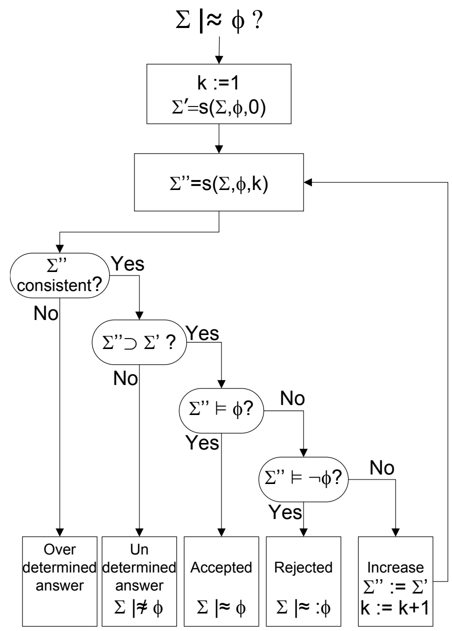


**Fig. 1.** Linear Extension Strategy.

track of the extension chain. The disadvantage of the linear strategy is that it may result in too many 'undetermined' or 'overdetermined' answers when the selection function picks the wrong sequence of monotonically increasing subsets.

In the case of $s(\Sigma, \phi, k)$ being inconsistent, we can refine the procedure from figure 1 with a backtracking step, which tries to reduce $s(\Sigma, \phi, k)$ to a set that still extends $s(\Sigma, \phi, k-1)$, but that is still consistent. This would reduce the number of overdetermined answers, and hence improve the linear extension strategy. We call this procedure *overdetermined processing*(ODP). ODP introduces a degree of non-determinism: selecting different maximal consistent subsets of $s(\Sigma, \phi, k)$ may yield different answers to the query $\Sigma \mathrel{|\!\approx} \phi$. An easy solution to overdetermined processing is to return the first maximal consistent subset (FMC) of $s(\Sigma, \phi, k)$, based on certain search procedure. Query answers which are obtained by this procedure are still meaningful, because they are supported by a consistent subset of the ontology. However, it does not always provide intuitive answers because it depends on the search procedure of maximal consistent subset in overdetermined processing. A natural search procedure is to perform breadth-first search among the subsets of $s(\Sigma, \phi, k)$ in decreasing cardinality, until we find the first (and hence maximal) consistent subset.

### 2.2 Syntactic Selection Functions

Direct relevance between two *formulas* is defined as a binary relation on formulas: $\mathcal{R} \subseteq \mathbf{L} \times \mathbf{L}$. Given any direct relevance relation $\mathcal{R}$, we can extend it to a relation $\mathcal{R}^+$ on a formula and a *formula set*, i.e. $\mathcal{R}^+ \subseteq \mathbf{L} \times \mathcal{P}(\mathbf{L})$, as follows:

$$\langle \phi, \Sigma \rangle \in \mathcal{R}^+ \text{ iff } \exists \psi \in \Sigma \text{ such that } \langle \phi, \psi \rangle \in \mathcal{R}.$$

In other words, a formula $\phi$ is relevant to a formula set $\Sigma$ iff there exists a formula $\psi \in \Sigma$ such that $\phi$ and $\psi$ are relevant. Two formulas $\phi, \phi'$ are $k$-relevant with respect to a formula set $\Sigma$ iff there exist formulas $\psi_0, \ldots \psi_{k+1} \in \Sigma$ such that $\phi = \psi_0$ $\psi_{k+1} = \phi'$ and all $\psi_i$ and $\psi_{i+1}$ are directly relevant.

We can use such a relevance relation to define a selection function $s$ as follows:

$s(\Sigma, \phi, 0) = \emptyset$
$s(\Sigma, \phi, 1) = \{\psi \in \Sigma | \phi \text{ and } \psi \text{ are directly relevant}\}$
$s(\Sigma, \phi, k) = \{\psi \in \Sigma | \psi \text{ is directly relevant to } s(\Sigma, \phi, k-1)\} \; for \; k > 1$

There are various ways to define a syntactic relevance $\mathcal{R}$ between two formulas in an ontology. Given a formula $\phi$, we use $I(\phi), C(\phi), R(\phi)$ to denote the sets of individual names, concept names, and relation names that appear in $\phi$ respectively. In [12], we proposed a direct relevance which considers the presence of a common concept/role/individual name in two formulas: two formulas $\phi$ and $\psi$ are directly syntactically relevant, written $\mathcal{R}_{SynRel}(\phi, \psi)$, iff there is a common name which appears in both formulas.

In [11, 12], we provided a detailed evaluation of the syntactic relevance approach by applying it to several inconsistent ontologies. The tests show that the syntactic relevance approach can obtain intuitive results in most cases for reasoning with inconsistent ontologies. The reason for this is that syntactic relevance mimics our intuition that real-world truth is (generally) preserved best by the argument with the shortest number of steps; and whatever process our intuitive reasoning uses, it is very likely that it would somehow privilege just these shortest path arguments[2]. However, as we will see, the problem is that the syntactic relevance approach requires that the syntactic encoding of the ontology by knowledge engineers correctly represents their intuitive understandings of the knowledge.

**Example:** A simple example where syntactic relevance works very well is the traditional penguin example in which birds are specified as flying animals and penguins are specified as birds which cannot fly. In this example, the reasoning path from *penguin* to $\neg fly$ is shorter than that from *penguin* to $fly$:

$penguin \sqsubseteq \neg fly, \quad penguin \sqsubseteq bird \sqsubseteq fly.$

**Example:** However, the syntactic relevance approach does not work very well on the MadCow example[3], in which Cows are specified as vegetarians whereas MadCows are specified as Cows which eats brains of sheep (and hence are not

[2] Thanks to Norman Gray, for pointing this out in personal communication.
[3] The Mad Cow ontology is used in OilEd tutorials

vegetarians). Under the syntactic relevance selection functions, the reasoner returns the 'accepted' answer to the query 'is the_mad_cow a vegetarian?'. This counter-intuitive answer results from the weakness of the syntactic relevance approach, because it always prefers a shorter relevance path when a conflict occurs. In the MadCow example, the path *'mad cow - cow - vegetarian'* is shorter than the path *'mad cow - eat brain - eat bodypart - sheep are animals - eat animal - not vegetarian'*. Therefore, the syntactic relevance-based selection function finds a consistent sub-theory by simply ignoring the fact 'sheep are animals'.

### 2.3 Pro's and Cons of syntactic relevance

**Empirically good results:** In [11, 12], we provided a detailed evaluation of the syntactic relevance approach by applying it to several inconsistent ontologies. The tests show that the syntactic relevance approach can obtain intuitive results in most cases for reasoning with inconsistent ontologies.

**Sensitive to syntactic encoding:** As shown above, the syntactic relevance approach is very dependent on the particular syntactic encoding that was chosen for the knowledge, since it selects short reasoning paths over longer ones. This works apparently works well in many cases (as shown in [11, 12]), but it is not hard to think of natural examples where the shortest reasoning chain is not the correct one to follow.

**Often needs a backtracking step:** Because of the "fan out" behaviour of the syntactic selection function, the relevance set will grow very quickly, and will become very large after a small number of iterations. A very large relevance set is in danger of becoming inconsistent itself, causing the system to need the backtracking step that we called "overdetermined processing".

**Backtracking is blind:** To make matters worse, the backtracking step of the syntactic approach is essentially blind. It is hard to think of ways to make this backtracking more involved, based only on syntactic features.

## 3 Semantic Selection Functions

A wide space of semantic relevance measures exist, varying from Wordnet distance [2], to the co-occurrence of concepts in search engines like Google [5, 4]. In this paper, we will use the latter, since we want to take advantage of the vast knowledge on the Web that is implicitly encoded in search engines. In this way, we can obtain light-weight semantics for selection functions.

The basic assumption here is that the more frequently two concepts appear in the same web page, the more semantically close they are, because most web pages are meaningful texts. Therefore, information provided by a search engine can be used to measure semantic relevance among concepts.

### 3.1 General properties of semantic relevance

Semantic relevance is considered as the reverse relation of semantic dissimilarity: the more semantically relevant two concepts are, the smaller the distance between them. Assuming that both relevance and distance are taken from the [0,1] interval, this boils down to $Similarity(x, y) = 1 - Distance(x, y)$[4].

To use semantic dissimilarity for reasoning with inconsistent ontologies, we define the dissimilarity measure between two formulas in terms of the dissimilarity measure between two concepts/roles/individuals from the two formulas. Moreover, in the following we consider only concept names $C(\phi)$ as the symbol set of a formula $\phi$ to simplify the formal definitions. However, note that the definitions can be easily generalised into ones in which the symbol sets contain also roles and individuals. We use $SD(\phi, \psi)$ to denote the semantic distance between two formulas. We expect the semantic distance between two formulas $SD(\phi, \psi)$ to satisfy the following intuitive properties:

**Range** The semantic distance is a real number between 0 and 1: $0 \leq SD(\phi, \psi) \leq 1$ for any $\phi$ and $\psi$.

**Reflexivity** Any formula is always semantically closest to itself: $SD(\phi, \phi) = 0$ for any $\phi$.

**Symmetry** The semantic distance between two formulas is symmetric: $SD(\phi, \psi) = SD(\psi, \phi)$ for any $\phi$ and $\psi$.

**Maximum distance** If all symbols in a formula are semantically most-dissimilar from any symbol of another formula, then the two formulas are totally dissimilar: if $SD(C_i, C_j) = 1$ for all $C_i \in C(\phi)$ and $C_j \in C(\psi)$, then $SD(\phi, \psi) = 1$.

**Intermediate values** If some symbols are shared between two formulas, and some symbols are semantically dissimilar, the semantic distance between the two formulas is neither minimal nor maximal: If $C(\phi) \cap C(\psi) \neq \emptyset$ and $C(\phi) \not\subseteq C(\psi)$ and $C(\psi) \not\subseteq C(\phi)$ then $0 < SD(\phi, \psi) < 1$.

### 3.2 Google distance as semantic relevance

In [5, 4], the Google Distance is introduced to measure the co-occurrence of two keywords on the Web. Normalised Google Distance (NGD) is introduced to measure the semantic distance between two concepts by the following definition:

**Definition 1 (Normalised Google Distance [4]).**

$$NGD(x, y) = \frac{max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - min\{\log f(x), \log f(y)\}}$$

*where $f(x)$ is the number of Google hits for the search term $x$, $f(y)$ is the number of Google hits for the search term $y$, $f(x, y)$ is the number of Google hits for the tuple of search terms $x$ and $y$, and $M$ is the number of web pages indexed by Google.*

$NGD(x, y)$ can be understood intuitively as the symmetric conditional probability of co-occurrence of the search terms $x$ and $y$. $NGD(x, y)$ is defined between two search items $x$ and $y$. Simple ways to extend this to measure the semantic

[4] In the following we use the terminologies *semantic dissimilarity* and *semantic distance* interchangeably.

distance between two formulas are to take either the minimal, the maximal or the average NGD values between two concepts (or roles, or individuals) which appear in two formulas as follows:

$$SD_{min}(\phi,\psi) = min\{NGD(C_i,C_j)|C_i \in C(\phi) \text{ } and \text{ } C_j \in C(\psi)\}$$
$$SD_{max}(\phi,\psi) = max\{NGD(C_i,C_j)|C_i \in C(\phi) \text{ } and \text{ } C_j \in C(\psi)\}$$
$$SD_{ave}(\phi,\psi) = \frac{sum\{NGD(C_i,C_j)|C_i \in C(\phi) \text{ } and \text{ } C_j \in C(\psi)\}}{(|C(\phi)| * |C(\psi)|)}$$

where $|C(\phi)|$ means the cardinality of $C(\phi)$. However, it is easy to see that $SD_{min}$ and $SD_{max}$ do not satisfy the Intermediate Values property, and $SD_{ave}$ does not satisfy Reflexivity.

We therefore propose a semantic distance which is measured by the ratio of the summed distance of the difference between two formulas to the maximal distance between two formulas:

**Definition 2 (Semantic Distance).**

$$SD(\phi,\psi) = \frac{sum\{NGD(C_i,C_j)|C_i \in C(\phi)\backslash C(\psi), C_j \in C(\psi)\backslash C(\phi)\}}{(|C(\phi)| * |C(\psi)|)}$$

The intuition behind this definition is to sum the semantic distances between all terms that are not shared between the two formulae, but these must be normalised (divided by the maximum distance possible) to bring the value back to the [0,1] interval. It is easy to prove the following:

**Proposition 1.** *The semantic distance $SD(\phi,\psi)$ satisfies the properties Range, Reflexivity, Symmetry, Maximum Distance, and Intermediate Values.*

Using the semantic distance defined above, the obvious way to define a relevance relation for selection functions in reasoning with inconsistent ontologies is to take the semantically closest formulas as directly relevant:

$$\langle\phi,\psi\rangle \in \mathcal{R}_{sd} \text{ iff} \neg\exists\psi' \in \Sigma : SD(\phi,\psi') < SD(\phi,\psi).$$

(i.e. there exist no other formulas in the ontology that is semantically closer)

Given this semantic relevance relation, we now need to define a selection function. In the syntactic approach of the previous section, we used the query formula as the starting point for the selection function. We can define a similar selection function $s$ in terms of the semantic relevance relation $\mathcal{R}_{sd}$. Namely, the newly defined selection function will track along the concept hierarchy in an ontology and always add to the selected set the closest formulas which have not yet been selected[5].

[5] It is easy to see the definition about $SD(\phi,\psi)$ is easily extended into a definition about $SD(\phi,C)$, where $\phi,\psi$ are formulas, and $C$ is a concept. Moreover, it is easy to see that $SD(\phi_1,C) < SD(\phi_2,C)$ iff $NGD(D_1,C) < NGD(D_2,C)$ for any $\phi_1$ is of the form $C_1 \sqsubseteq D_1$ and any $\phi_2$ is of the form $C_1 \sqsubseteq D_2$ where $C, C_1, D_1$ and $D_2$ are different concepts.

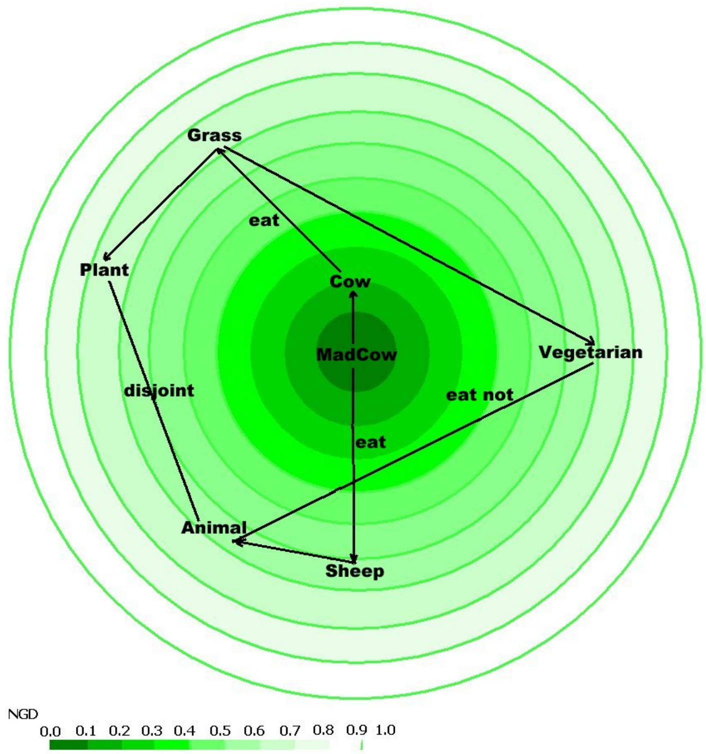


**Fig. 2.** NGD and MadCow Queries

**Example:** Figure 2 shows how the semantic distance is used to obtain intuitive answers on the MadCow ontology (where the syntactic distance failed). By calculation of the Normalised Google Distance, we know that

$$NGD(MadCow, Grass) = 0.722911, \ \ NGD(MadCow, Sheep) = 0.612001.$$

Hence, the semantic distance between MadCow and Sheep is shorter than the semantic distance between MadCow and Grass (even though their syntactic distance is larger). Because of this, the reasoning path between *MadCow* and *Sheep* is preferred to the reasoning path between *MadCow* and *Grass*. Thus, we obtain the intuitive answer that *MadCow are not Vegetarians* instead of the previously obtained counter-intuitive answer that *MadCow are Vegetarians*. The intuition here is that although syntactically, the *MadCow* - *Sheep* path is the longer of the two, the accumulated semantic distance on this syntactically longer path is still shorter than the semantic distance on the syntactically short *MadCow* - *Grass* path.

### 3.3 Pro's and Cons of semantic relevance

Although empirical findings will only be discussed in section 5, we can already establish some of the advantages and disadvantages of the semantic approach to relevance.

**Slower fan out behaviour:** As is clear from the definition the growth of a relevance based on semantic distance is much slower than one based on syntactic relevance. In fact, at each step the semantic relevance set grows by a single formula (barring the exceptional case when some formulas share the same distance to the query).

**Almost never needs a backtracking step:** This slower growth of semantic relevance means that it will also hardly ever need a backtracking step, since the relevance set is unlikely to become "too large" and inconsistent.

**Expensive to compute:** Again by inspecting the definition, it is clear that computing the semantic relevance is expensive: it requires to know the semantic distance between the query and *every* formula $\psi$ in the theory $\Sigma$. Furthermore, this must be done again for every new query concept $C_1$. With realistic modern ontologies often at a size of $O(10^5)$ concepts, and a computation time in the order of 0.2 secs for a single NGD-value, this would add a prohibitive cost to each query[6].

## 4 Mixed Approach

The picture that emerges from the pro's and cons in sections 2.3 and 3.3 is syntactic relevance is cheap to compute, but grows too quickly and then has to rely on a blind backtracking step, while semantic relevance has controlled growth, with no need for backtracking, but is expensive to compute.

In this section, we will propose a *mixed* approach which combine the advantages of both: we will use a syntactic-relevance selection function to grow the selection set cheaply, but we will use semantic relevance to improve the backtracking step. Instead of picking the first maximal consistent subset through a blind breadth-first descent, we can prune semantically less relevant paths to obtain a consistent set. This is done by removing the most dissimilar formulas from the set $s(\Sigma, \phi, k) - s(\Sigma, \phi, k-1)$ first, until we find a consistent set such that the query $\phi$ can be proved or disproved.

**Example:** Taking the same example of the MadCow ontology above, we can see from Figure 2 that the path between *MadCow* and *Grass* can be pruned first, rather than pruning the path between *MadCow* and *Sheep*, because the NGD between $MadCow$ and *Sheep* is smaller than the NGD between $MadCow$ and $Grass$. Thus, the path *MadCow* - *Grass* (which lead to the counter-intuitive conclusion that *MadCow are vegetarians*) is pruned first.

We call this *overdetermined processing (ODP) using path pruning with Google distance*. While syntactic overdetermined processing (from section 2.1) can be

[6] Although some of this can be amortised over multiple queries by caching parts of the values that make up the NGD (definition 1).

seen as a blind breadth-first search, semantic-relevance ODP can be seen as a hill-climbing procedure, with the semantic distance as the heuristic.

**Possible loss of completeness and soundness.** Notice that semantic backtracking is not guaranteed to yield a *maximal* consistent subset. Consequently, the completeness of the algorithm may be affected, since we might have removed too many formulas from the relevance set in our attempt to restore consistency, thereby loosing the required implication to obtain the intuitive answer. Furthermore, it is possible that the semantic backtracking might lead to the *wrong* consistent subset, one supporting $\phi$ where $\neg\phi$ would have been the intuitive answer, or vice versa. In our experiment in section 5 we will find that indeed the completeness drops (as expected), but not by very much, while the unsoundness does not increase at all (making us belief that SD is a good heuristic for the hill-climbing search towards a consistent subset).

**Cutting levels in ODP.** Finally, the semantic distance provides the possibility for adjustable behaviour of the backtracking increments that are taken in the overdetermined processing phase. We introduce a cutting level $\alpha$ ($0 \leq \alpha \leq 1$), and instead of only pruning the semantically least relevant paths one by one until we obtain a consistent subset, we now prune in one step all formulas whose distance to the query is higher than $\alpha$. In this way, $\alpha$ plays the role of a threshold, so that the processing can be sped up by pruning in a single step all those formulas which do not meet the relevance threshold. This might of course increase the amount of undetermined answers (since we may have overpruned), but it allows us to make a tradeoff between the amount of undetermined answers and the time performance. In Section 5 we will report an experiment in which this tradeoff obtains a 500% efficiency gain in exchange for only a 15.7% increase in undetermined answers.

## 5 Implementation and Experiments

We have implemented these definitions and algorithms in PION (Processing Inconsistent ONtologies)[7]. In this section, we report several experiments on reasoning with inconsistent ontologies using the selection functions introduced above.

### 5.1 Data

As already observed before, many ontologies on the Semantic Web (e.g. those indexed by Swoogle[8]) do not contain explicit inconsistencies. This would make it hard to obtain test-data for running our experiments and indeed, it would question the need for our inconsistency reasoning methods in the first place. The following brief analysis shows that under the surface, the situation is different.

Disjointness constraints between classes in an ontology are necessary for a suitable formalisation of a conceptualisation, and are required to draw the required inferences in tasks such as search, navigation, visualisation, service matching, etc [19]. However, as shown by [17] and [16], knowledge engineers often

[7] available for download at `wasp.cs.vu.nl/sekt/pion`

[8] `swoogle.umbc.edu`

neglect to add disjointness statements to their ontologies, simply because they are not aware of the fact that classes which are not explicitly declared to be disjoint will be considered as potentially overlapping. Furthermore, an experiment by Völker and her colleagues in [19] showed that when prompted to add disjointness statements, human experts are very prone to introducing inadvertent inconsistencies. Since we will take the ontologies that resulted from that experiment as our dataset, we will describe that experiment in some detail.

The experiment in [19], takes as its starting point a subset of the PROTON Ontology[9]. The selected subset of PROTON contains 266 classes, 77 object properties, 34 data-type properties and 1388 siblings. Each concept pair was randomly assigned to 6 different people - 3 from a group of professional "ontologists", and 3 from a group of students without profound knowledge in ontological engineering. Each of the annotators was given between 385 and 406 concept pairs along with a natural language descriptions of the classes whenever those were available, and were asked to annotate each concept pair as "disjoint", "overlapping" or "unknown". Two enriched versions of the ontology were then constructed by adding those disjointness statements that were agreed upon by 100% of the experts and of the students respectively. We will call these the `experts` and the `students` ontologies respectively. These two ontologies were both inconsistent. For example, the `students` ontology alone already contained some 24 unsatisfiable concepts. Even more telling is the following example:

**Example:** 100 percent of students and experts (!) agree on the following axioms, which are, however inconsistent:

$$\begin{aligned} Reservoir &\sqsubseteq Lake \\ Lake &\sqsubseteq WaterRegion \\ Reservoir &\sqsubseteq HydrographicStructure \\ HydrographicStructure &\sqsubseteq Facility \\ Disjoint(&WaterRegion, Facility) \end{aligned}$$

This case shows that inconsistency and incoherence occurs much more easily than what is often expected. Interestingly enough, this problem would be handled by our semantic relevance approach. Normalised Google Distance tells us that *Lake* and *WaterRegion* are more semantically relevant than *Facility* and *HydrographicStructure* to *Reservoir*. Thus, using the semantic relevance based selection function, we would conclude that *Reservoir* is a *WaterRegion*.

The essence of all this is that although the original ontology did not contain inconsistencies (due to lack of disjointness statements), the inconsistencies arise as soon as human knowledge engineers are asked to add explicit disjointness statements to the best of their capabilities. Thus, the resulting ontologies contain "natural" inconsistencies. This makes the resulting set of inconsistent ontologies a realistic data-set for our experiments.

[9] `proton.semanticweb.org/`

| Ontology | Method | $\alpha$ | Query | IA(IA rate) | CA | RA | CIA | ICRate(%) | Time | TRatio |
|---|---|---|---|---|---|---|---|---|---|---|
| `experts` | FMC | n/a | 529 | 266 (50%) | 219 | 32 | 12 | 91.68 | 114.63 | n/a |
| `experts` | SD | 0.85 | 529 | 246 (47%) | 238 | 32 | 13 | 91.49 | 54.28 | 2.11 |
| `experts` | SD | 0.80 | 529 | 239 (45%) | 246 | 32 | 12 | 91.68 | 39.96 | 2.87 |
| `experts` | SD | 0.75 | 529 | 225 (43%) | 260 | 32 | 12 | 91.68 | 22.37 | 5.12 |
| `students` | FMC | n/a | 529 | 234 (44%) | 249 | 33 | 13 | 91.30 | 45.05 | n/a |
| `students` | SD | 1.00 | 529 | 189 (36%) | 309 | 22 | 9 | 94.14 | 28.11 | 1.62 |

IA = Intended Answers, CA = Cautious Answers, RA = Reckless Answers, CIA = Counter-Intuitive Answers, IA Rate = Intended Answers(%), IC Rate = IA+CA(%), FMC = First Maximal Consistent subset, SD = Semantic Distance, $\alpha$=Threshold, Time = Time Cost per Query (seconds), TRatio = TimeCost(FMC)/TimeCost(SD)

**Fig. 3.** PION test results by using FMC or SD for overdetermined processing

### 5.2 Tests

**Goal.** Given that the mixed approach (using syntactic relevance for growing the relevant set, and using semantic relevance for backtracking, possibly using $\alpha$-cuts) seems to be the best alternative to the purely syntactic approach of our earlier work, our experiment is aimed at (1) finding out the quality of the answers generated by the mixed approach, and (2) finding out the quality/cost trade-offs that can be obtained by varying the $\alpha$-levels.

**Test Queries and Answers.** We created 529 subsumption queries randomly, and obtained PION's answers of these queries with backtracking done either blindly (First Maximal Consistent Subset, FMC), or via the semantic distance (SD). We compared these answers against a hand-crafted Gold Standard that contained the humanly-judged correct answer for all of these 529 queries. For each query, the answer given by PION can be classified in one of the following categories, based on the difference with the intuitive answer in the Gold Standard:

**Intended Answer**: PION's answer is the same as the intuitive answer from the Gold Standard.

**Counter-intuitive Answer**: PION's answer is opposite to the intuitive answer, i.e. the intuitive answer is "accepted" whereas PION's answer is "rejected", or vice versa.

**Cautious Answer**: The intuitive answer is "accepted" or "rejected", but PION's answer is "undetermined".

**Reckless Answer**: PION's answer is "accepted" or "rejected" while the intuitive answer is "undetermined".

Obviously, one would like to maximise the Intended Answers, and minimise the Reckless and Counter-intuitive Answers. Furthermore, we introduced different $\alpha$-thresholds in the overdetermined processing to see how the tradeoff between the quality of query-answers and the time performance is effected by different cutting levels.

### 5.3 Results

Our results obtained by running PION with the data and the tests described above are shown in Figure 3. The first 4 rows show experiments on the `experts` ontology, the final 2 rows on the `students` ontology. In all cases, we use syntactic relevance for growing the relevance set until an answer can be found, but they differ on what happens when the relevance set becomes inconsistent, and backtracking is required. On the first line (labelled FMC, for First Maximal Consistent subset), the backtracking is done blindly, on the other lines, backtracking is guided by the semantic distance function, at different $\alpha$-levels (i.e. with different sizes of the backtracking steps; smaller values for $\alpha$, i.e. lower thresholds, means that more formulas are removed during backtracking). Not listed in the table is the fact that among the 529 queries, 414 (i.e. 78%) resulted in relevance sets that became inconsistent before the query could be answered meaningfully, hence they needed a backtracking phase.

**Answer quality.** The tables shows that when switching from syntactic backtracking (labelled FMC) to semantic backtracking (labelled SD) the intended answer (IA) rate does indeed drop, as predicted in section 4. Furthermore, the IA-rate declines slowly with decreasing $\alpha$-levels. Similarly, the cautious answer rate increases slowly with decreasing $\alpha$-levels. This is again as expected: larger backtracking steps are more likely to remove too many formulas from the relevance set, hence potentially making the relevance set too small. Or put another way: the hill-climbing search performed in the ODP phase is aiming to get close to a maximal consistent subset, but larger hill-climbing steps make it harder to end up close to such a set, because of possible overpruning.

The combined IC-rate (combining intended and cautious answers, i.e. those answers that are not incorrect, but possibly incomplete), stays constant across between FMC and SD, and across all $\alpha$-levels. It is important to note that the numbers of reckless and counter-intuitive answers remains constant. This means that although the semantically guided large-step reductions (at low $\alpha$-levels) do of course remove formulas, they do not remove the wrong formulas, which could have potentially lead to reckless or counter-intuitive answers.

Summarising, *when switching from FMC to SD, and with decreasing $\alpha$-levels, the completeness of the algorithm (IA Rate) gradually declines, while the soundness of the algorithm (IC rate) stays constant.*

**Cost/Quality trade-offs.** Although these findings on the answer quality are reassuring (the semantic backtracking doesn't damage the quality), they are not by themselves a reason to prefer semantic backtracking over syntactic backtracking. The strong point of the semantic backtracking becomes clear when we look at the computational costs of syntactic and semantic backtracking, particularly in the light of the answer quality.

Above, we have seen that the answer quality only degrades very gradually with decreasing $\alpha$-levels. The final two columns of table 3 however show that the answer *costs* reduce dramatically when switching from syntactic to semantic backtracking, and that they drop further with decreasing $\alpha$-levels. The absolute computation time is more than halved when switching from FMC to SD ($\alpha$ =

0.85), and is again more than halved when dropping $\alpha$ from 0.85 to 0.75, leading to an overall efficiency gain of a factor of 5. Of course, this efficiency is gained at the cost of some loss of quality, but this loss of quality (the drop in completeness, the IA rate) is very modest: the twofold efficiency gain at $\alpha = 0.85$ is gained at a price of a drop of only 3 percentage points in completeness, and the fivefold efficiency gain at $\alpha = 0.75$ is gained at a price of a drop of only 7 percentage points in completeness. Summarising, *semantic backtracking with cut-off levels yields a very attractive cost/quality trade-off between costs in terms of run-time, and the quality in terms of soundness and completeness of the answers.*

## 6 Discussion and Conclusions

Research from a number of different areas is relevant to the current work. Semantic distances and similarity measures have been widely used in computational linguistics [2, 14] and ontology engineering [8, 15]. [7] proposes the use of a Google-based similarity measure to weigh approximate ontology matches. Our research in this paper is the first attempt to introduce the Google Distance for reasoning with inconsistent ontologies. In essence we are using the implicit knowledge hidden in the Web for explicit reasoning purposes.

The main contributions of this paper are: a) we investigated how a semantic relevance-based selection function can be developed by using information provided by a search engine, in particular, by using the Normalized Google Distance; b) we provided variants of backtracking strategies for reasoning with inconsistent ontologies, and c) we showed that semantic distances can be used for handling large scale ontologies through a tradeoff between run-time and the degree of incompleteness of the algorithm.

In our experiment we applied our PION implementation to realistic test data. The experiment used a high-quality ontology that became inconsistent after adding disjointness statements that had the full support of a group of experts. The test showed that the run-time of informed semantic backtracking is much better than that of blind syntactic backtracking, while the quality remains comparable. Furthermore the semantic approach can be parametrised so as to stepwise further improve the run-time with only a very small drop in quality.

Clearly, our experiments should be repeated on many different ontologies in order to see how generalisable our results are. We are now developing a benchmark system for reasoning with inconsistent ontologies, by which various approaches and selection functions can be tested with different application scenarios on much larger ontology data. This is an inherently difficult task, because existing ontologies will often need to be enriched by making disjointness statements explicit before they can be used as test data. Furthermore, a Gold Standard of intuitive answers can often only be created by hand. These high costs experiment-construction costs also justify why we did not run more experiments in the scope of this paper.

One of the future tasks is to make the NGD (Normalized Google Distance) component well integrated with the architecture of PION, so that the NGD values can be dynamically obtained at run time, rather than as the pre-loaded libraries, as it is done in the present implementation.

**Acknowledgements:** The work reported in this paper was partially supported by the EU-funded projects SEKT, KnowledgeWeb, and LarKC. We thank Annette ten Teije for useful clarifying discussions and Johanna Völker for providing the inconsistent PROTON ontologies.

## References

1. N. Belnap. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logic*, pages 8–37, Dordrecht, 1977. Reidel.
2. A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet*. Pittsburgh, PA., 2001.
3. S. Chopra, R. Parikh, and R. Wassermann. Approximate belief revision- prelimininary report. *Journal of IGPL*, 2000.
4. R. Cilibrasi and P. Vitany. The Google similarity distance. *IEEE/ACM Transactions on Knowledge and Data Engineering*, 19:3:370–383, 2007.
5. R. Cilibrasi and P. Vitanyi. Automatic meaning discovery using Google. Technical report, Centre for Mathematics and Computer Science, CWI, 2004.
6. M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing knowledge on the semantic web with watson. In *5th International EON Workshop at ISWC07*, 2007.
7. R. Gligorov, Z. Aleksovski, W. ten Kate, and F. van Harmelen. Using Google distance to weight approximate ontology matches. In *Proceedings of WWW 2007*, 2007.
8. P. Haase. *Semantic Technologies for Distributed Information Systems*. PhD thesis at the Universität Karlsruhe, 2006.
9. P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *Proceedings of ISWC05*, 2005.
10. A. Hameed, A. Preece, and D. Sleeman. Ontology reconciliation. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 231–250. Springer Verlag, 2003.
11. Z. Huang and F. van Harmelen. Reasoning with inconsistent ontologies: Evaluation. Project Report D3.4.2, SEKT, 2006.
12. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *Proceedings of IJCAI'05*, 2005.
13. Z. Huang, F. van Harmelen, A. ten Teije, P. Groot, and C. Visser. Reasoning with inconsistent ontologies: a general framework. Project Report D3.4.1, SEKT, 2004.
14. D. Lin. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning, Madison, Wisconsin, July*, 1998.
15. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of EKAW-2002*, pages 251–263, 2002.
16. A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. Owl pizzas: Practical experience of teaching owl-dl - common errors and common patterns. In *Proceedings of EKAW 2004*, pages 63–81, 2004.

17. S. Schlobach. Debugging and semantic clarification by pinpointing. In *ESWC*, pages 226–240, 2005.
18. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proceedings of IJCAI'03*, 2003.
19. J. Volker, D. Vrandecic, Y. Sure, and A. Hotho. Learning disjointness. In *Proceedings of ESWC2007*, 2007.