Avoiding Deceptive Annotations in the Semantic Web

Yihong Ding,* David Embley
Dept. of Computer Science
Brigham Young University
Provo, Utah, USA
(ding,embley)@cs.byu.edu

Ying Ding, Omair Shafiq
DERI
University of Innsbruck
Innsbruck, Austria
(ying.ding,omair.shafiq)@deri.org

ABSTRACT

Deceptive annotations are becoming an important problem as more and more people start to tag documents, and the problem has become an argument to against the Semantic Web. Skeptics believe that developers make mistakes when annotating documents, and developers may even abuse annotations from time to time. Due to the difficulty of detecting and resolving deceptive tags, these skeptics openly wonder whether semantic annotations may bring more trouble than benefit. In this paper we present a deception avoidance resolution method. By adding personal specifications about ontology concepts through instance recognition semantics, Semantic Web users can avoid being deceived by improperly annotated data. At the same time, our deception avoidance strategy also passively discourages annotators from falsely tagging documents by decreasing the profit they can gain from deceptive annotations. Finally, our deception avoidance mechanism still preserves the right to annotate text without restriction.

1. INTRODUCTION

Deceptive annotations, or deceptive tags, are becoming more and more of a problem as people start to tag their documents. The problem has, in fact, become an argument to against the Semantic Web. As an example, at a recent conference in Boston, Peter Norvig, the Google Director of Search and an AAAI Fellow, asked Tim Berners-Lee, the inventor of the Web and the current director of W3C, a question about deception in the Semantic Web [4]. Norvig said, "We deal every day with people who try to rank higher in the results and then try to sell someone Viagra when that's not what they are looking for. With less human oversight with the Semantic Web, we are worried about it being easier to be deceptive." In this question, Norvig reveals one of his concerns about the Semantic Web. Without question, Internet deception is a severe problem. Particularly in the Semantic Web, annotations are easily abused. If we cannot resolve deceptive annotations, we may have negative experiences with Semantic Web applications due to the unsure quality of annotated data.

In general, there are two opposing opinions to this problem.[1] Some people believe that deceptive annotation is a type of cheating. In their definition, deceptive annotations are false claims whose purpose is to mislead. Advocates of total freedom on the Internet, however, suggests that everybody has a right to say and write whatever is on their mind. So, in essence no annotations are "deceptive," but are only abnormal.

In the Semantic Web, *deceptive annotations* are the annotations with instances that deviate from their commonly expected meanings. For example, if "UTAH" is annotated as a $NATION$, this annotation is deceptive because a $NATION$ is commonly understood as an independent country in the world, which Utah is not.

At the same time, however, we must not prohibit the freedom of people to annotate as they wish. Things can change and new knowledge is discovered from time to time. For example, annotating "Montenegro" as a $NATION$ before June 3, 2006 would have been deceptive. But it is no longer deceptive after June 3, 2006, when Montenegro declared its independence. Moreover, people should have the freedom to annotate a document according to their own understanding even if it is seen as deceptive by others. For example, a Montenegro independence movement member may annotate "Montenegro" as a $NATION$ even before June 3, 2006. This was what the person believed and expected although it would certainly have been a deceptive annotation as viewed by others. To the end that the web is designed to be an open and free space, a resolution to the deceptive annotation problem should not override the freedom of tagging.

There are three strategies we can apply to solve the deceptive annotation problem: deception protection, deception detection, or deception avoidance. A deception protection strategy would allow only trusted authorities to annotate all web pages and would encrypt annotations so that no one can abuse them. Based on current Internet security technologies, we can believe that the deceptive annotation problem can be solved by deception protection methods. A problem with this resolution, however, is that it generally dismisses the right of individual web developers to annotate their own documents themselves.

A deception detection strategy would check the correctness of mappings between annotated data and their annota-

*Written mainly while this author was on an extended visit at DERI Innsbruck.

[1] http://www.bloghop.com/tagview.htm?itemid=deceptive

SAAW '06 Athens, GA, USA

tions based on formal definitions of rules in ontologies. Such a process is usually expensive to execute, however. For example, to check whether "Montenegro" is a $NATION$, a process must at least compare the annotating date to the independence date of Montenegro. Even worse, it could be very difficult to construct these rules and agree on them. Both defining rules as well as processing them would likely be costly. Researchers must first resolve all these sophisticate issues before we could really apply deception detection mechanisms.

In this paper, we present a deception avoidance strategy. Rather than detecting false annotations, the deception avoidance strategy avoids looking for potentially deceptive cases. Our method is based on two observations and assumptions: (1) users need not care about whether an annotation is deceptive unless they are interested in the annotation; and (2) if users are interested in an annotation, they can avoid being deceived by explicitly and clearly expressing their interests about the annotation. We proffer instance recognition semantics to allow Semantic Web users to specify their personal interests to avoid deceptive annotations. The degree of vulnerability to deceptive annotations depends on how precisely they have specified their instance recognition semantics in ontologies. Moreover, our deception avoidance strategy also passively discourages annotators from falsely tagging documents by decreasing the profit they can gain from deceptive annotations. At the same time, our deception avoidance method still preserves the right of people to annotate their documents freely.

To explain how our strategy works, we briefly introduce instance recognition semantics in Section 2. In Section 3, we show how we use instance recognition semantics in our deception avoidance strategy. Finally, we summarize the paper in Section 4.

2. INSTANCE RECOGNITION SEMANTICS

Instance recognition semantics, which can also be called *instance semantics recognizers* (ISR),[2] are formal specifications that identify instances of a concept C in ordinary text. The text may be unstructured, semi-structured, or fully structured. For Semantic Web applications, the concept C should be a lexical element of a formal ontology (e.g. concepts such as *date*, *time*, *place*, *location*, *name*, *telephone number*, *email address*, various weights and measures, etc.). Thus, instance recognition semantics of an ontology concept (e.g. $Telephone\ Number$) interpret instances in a text fragment (e.g. the contact number in "Call me at 222-1234.") to have the intensional meaning of the defined concept (e.g. $Telephone\ Number$).

Figure 1 shows a partial ISR declaration we have used in an apartment-rental domain ontology for the concept $BedroomCount$.[3] Although recognition patterns can be expressed variously in different syntaxes, in our study we have used Perl-style regular expressions. In general, an ISR declaration includes defined recognition patterns and auxiliary filtering specifications. We specify recognition patterns in an *external representation* clause. In Figure 1 we specify that any legal instantiation of $BedroomCount$ should be a string

```
BedroomCount
   external representation: [1-9]|\d|20
   left context phrase: \b
   right context phrase: .*r(oo)?ms?
   exception phrase: \s.*ba(th)?s?\b.*r(oo)?ms?
   context keyword: b(r|d)s? | bdrms? | bed(rooms?)?
   ...
end
```

Figure 1: Instance recognition semantics declarations for $BedroomCount$.

of digits representing numbers between 1 and 20. Defined auxiliary filtering specifications help to precisely identify an instance. In Figure 1, we declare the left immediate context (*left context phrase*) to be a legal word boundary and the right immediate context (*right context phrase*) to be the regular expression "r(oo)?ms?" with possibly several other words in between, e.g. "large room." The *exception phrase* excludes some negative phrases from the previously specified patterns, which is the *right context phrase* in our example. In our case, we exclude, for example, "bath room" to be a legal right context phrase. The *context keywords* are a carefully selected set of keywords that typically appear close to the concept locations. They are mainly for the purpose of improving the accuracy of automated semantic annotation processes. Although this example somehow looks complicated, many times ISR declarations can be as simple as a list of potential instances, such as a list of country names for the concept $NATION$.

ISR augmentations to ontologies help separate the work load between domain experts (who are individual annotators) and data-extraction engineers (who design and build data-extraction engine). This separation is key in our automatic deception avoidance mechanism. Because ISR rules are declarative, domain experts can create instance recognition rules for domain concepts without having to do any programming; and because ISR rules are embedded inside of ontologies, domain experts need not be concerned about mapping recognized concepts to domain ontologies.[4] These two properties of ISR rules enable domain experts to create and update their ISR declarations without the need to consult with data-extraction engineers. Since domain experts know their domain best, their ISR declarations can best protect domain integrity.

Using ISR declarations, domain experts implicitly "personalize" the meanings of specified ontology concepts. Here *personalize* means that domain experts cast the recognition of a generally defined concept to their own expectations. Figure 2 illustrates this idea with a simple example. Without ISR declarations, an arbitrary positive integer number could be a legal instantiation of the concept $BedroomCount$,[5] although in reality we rarely can find a single apartment with more than 4 or 5 bedrooms. With ISR declarations, we can restrict the instantiation of $BedroomCount$ to be between 3 and 4, perhaps because we need an apartment with at least three bedrooms and we do not anticipate ever needing more than four bedrooms. Therefore, our $BedroomCount$ with this ISR declaration becomes a spe-

[2]We avoid the acronym IRS (Internal Revenue Service) because instance recognition semantics are not tax collectors.
[3]The ontology can be found in the DEG web site: http://www.deg.byu.edu/

[4]Mapping concepts to domain ontologies is a major concern in current semantic annotation approaches [1, 2, 3].
[5]In theory there is no restriction why one cannot build a mansion with 100 bedrooms.

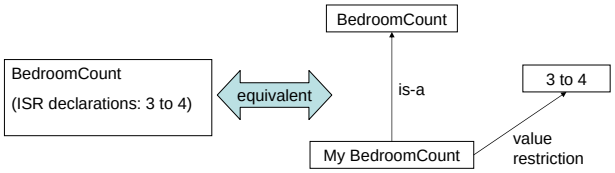

Figure 2: Concept with IRS declarations equivalent to declare a special subclass to itself.

cialization of the *BedroomCount* without an ISR declaration or with a different (more generalized) ISR declaration. Hence the meaning of the concept *BedroomCount* is personalized to our perspective. With personalized concepts, ontologies become personalized, augmented by personalized ISR declarations.

3. DECEPTION AVOIDANCE

Deceptive annotations are harmless if users are not interested in them. For example, if "Viagra" is falsely annotated as a *FOOD*, users will not be deceived unless they are looking for *FOOD*. Therefore, users can automatically avoid deceptive annotations in which they are not interested. Moreover, if users are interested in an annotation, they can avoid being deceived by explicitly and clearly expressing their interests about the annotation. For example, if users are looking for *FOOD*, and they have clearly specified that their *FOOD* consists of lists of breads, meats, and vegetables, they can also avoid being deceived by "Viagra" since it is not on their list. These two scenarios constitute the basis of our deception avoidance methodology.

By augmenting ISR declarations, ontologies become personalized ontologies. Therefore, any annotations that contradict specified personal interests can be automatically ignored. For example, the following house-rental advertisement is from a real online web site,[6] and we have intentionally annotated it deceptively with our apartment-rental ontology.

```
<BedroomCount>3.5</BedroomCount> Bed,
<BathroomCount>2.5</BathroomCount> Bath
House with <Feature>Pool</Feature>,
<Feature>Large LCD HDTV</Feature>,
<Feature>High speed internet</Feature>
```

By applying the ISR declarations in Figure 1, however, machines can avoid being deceived by these deceptive annotations because "3.5" is not recognized as a data instance of interest by the specified *external representation* for the concept *BedroomCount*. In this process, machines do not generate any logic rules from ontologies to detect the semantic meaning of this annotated data; nor do machines perform any domain identification methods to verify the application domain for this advertisement. Machines avoid this deceptive case simply because of the ISR declaration in the adopted ontology.

On the other hand, perhaps we begin to notice several $n.5$ bedroom counts. The following example is also from a real online advertisement,[7] and we have annotated it manually with our apartment-rental ontology.

```
<Feature>Large</Feature> <BedroomCount>2.5
</BedroomCount> room apartment 70 qm available
<AvailableDate>July 1</AvailableDate>
```

Although we still may not know what the meaning of a ".5 bedroom" is, somebody truly has expressed the number of bedrooms like this. We have two choices: either we can modify our *external representation* declaration so that it accepts $n.5$ as a legal representation for room numbers, or we can keep ignoring them and continue to treat them as deceptive annotations because we do not like $n.5$ bedrooms. Both choices are fine, and the decision totally depends on personal perspectives.

Using this same technique, we can resolve the problem that a deceiver falsely annotates "Viagra" as a *FOOD* in order to attract more readers to a Viagra-sales web page. This deception may not be easy to detect through ontology reasoning because Viagra is edible, which satisfies one of the crucial feature about *FOOD*. But we can avoid this problem by applying our deception avoidance method. Based on different conditions, there are two ways to avoid this deception. First, if users specify a list of *FOOD* items that does not contain Viagra, straightforwardly they avoid this deceptive web page based upon unmatched interests. Second, if users are open to trying new foods that they do not know, they can simply leave the *external representation* of their *FOOD* declaration blank, which means that they accept whatever is annotated as a *FOOD* to be *FOOD*. Then they will be deceived by this deceptive annotation the first time. But after they learn that this is a deception, they can avoid it by simply adding an *exception phrase* "Viagra" for their *external representation* about *FOOD*. Hence they would never be trapped in this deception again. This update avoids not only this deceptive web page, but also all the other web pages that play the same deceptive trick on readers.

In our deception avoidance method, we must emphasize that the vulnerability of users to deceptive annotations depends very much on how carefully users build and improve their ISR declarations. It is fair, however. Just like in any human society, humans who are too lazy to learn will be repeatedly deceived by the same trick. Only if they learn from previous experiences, i.e. only if they update their own ISR declarations by their experiences, can they avoid being deceived again. When we continually update our knowledge by our experiences, we become harder and harder to deceive. Hence our deception avoidance method is partly an incremental self-learning process.

Since our method does not depend on annotations, but rather on recognizers, our method preserves total freedom for annotators to tag whatever they want to any textual content. Our method is applied to the user side rather than the annotator side. While users have the power to avoiding what they believe to be deception, annotators can still annotate everything freely. For example, our method does not prohibit annotators from tagging "Viagra" to be a *FOOD*.

If our deception avoidance methods were used extensively on the web, deceptive annotators would find that they lose much more than they gain by deceptive annotations. For

[6]http://www.villas2000.com/frbvo/homes/3345.php. Checked August 6th, 2006.

[7]http://berlin.craigslist.org/apa/173491092.html. Checked August 6th, 2006.

example, the reason deceivers falsely annotate "Viagra" as a $FOOD$ is that they want to increase the hit rate of a web page. With our deception avoidance strategy, real food-seekers will soon learn that this is a deceptive web page and thus avoid visiting it any more. At the same time, real Viagra-seekers may look for annotations such as $MEDICINE$ rather than $FOOD$ because they do not think Viagra is a $FOOD$. Even if deceivers annotate "Viagra" simultaneously to be both $FOOD$ and $MEDICINE$, they still decreases their own opportunities to have their real customers because the thought that Viagra is not $FOOD$ overrides the thought that Viagra is both $FOOD$ and $MEDICINE$. Therefore, our mechanism not only provides an active deception avoidance method for users, but also becomes a passive deception avoidance strategy from an annotator's perspective.

4. CONCLUDING REMARKS

Deceptive annotations are becoming a severe problem as more and more people start to tag web data. Indeed it has been used as an argument against the realization of the Semantic Web. In this paper we presented a new deception avoidance resolution. By augmenting ontologies with ISR declarations, our method not only provides active deception avoidance for users, but may also passively decrease the rate of deception by reducing the chances that deceivers may obtain benefits from deceptive annotations. We expect that our work may lead to more attention being paid to this important and interesting research problem.

5. REFERENCES

[1] Y. Ding, D.W. Embley, and S.W. Liddle. Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In *Proceedings of the first Asian Semantic Web Conference (ASWC 2006), LNCS 4185*, pages 400–414, Beijing, China, September 2006.

[2] S. Handschuh, S. Staab, and F. Ciravegna. S-cream semi-automatic creation of metadata. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW-2002)*, pages 358–372, Madrid, Spain, October 2002.

[3] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2(1):49–79, December 2004.

[4] C. Lombardi. Google exec challenges Berners-Lee. http://news.zdnet.com/2100-9588_22-6095705.html.