**RC22979 (W0311-058) November 7, 2003**
**Computer Science**

# IBM Research Report

## The Semantics of Multiple Annotations

**Christopher A. Welty, J. William Murdock**
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

# The Semantics of Multiple Annotations

Christopher A. Welty and J. William Murdock

IBM Watson Research Center

{welty,murdockj}@watson.ibm.com

*Abstract.* In the context of a project to populate knowledge-bases from large-scale unstructured information, we have encountered numerous problems integrating the extracted information and the knowledge-base, due to mismatches in semantics between the two. Semantic integration in general is quite hard, however we have found that many of the integration problems stem from unclear or ambiguous semantics on the extraction side. We have begun to catalog and document these ambiguities and report here on one in particular: the use and intended meaning of multiple annotations on a span of text. Multiple annotations are a representation structure used in at least seven semantically distinct ways by existing information extraction systems. This overloading makes the representation ambiguous, and creates an obstacle for integration. We focus on these different semantic interpretations and propose a more expressive representation that allows their differences to be expressed.

## Introduction

Information extraction (IE) systems are designed primarily to provide shallow linguistic processing on large amounts of data, and the technology varies across a wide range of different techniques and tasks. One particular thread is a view of the product of IE as *annotations* on spans of text. These annotations are often understood to carry some semantics, such as the name of the type of the entity named in the text. In particular, *named entity extraction* or *entity detection* is the process of recognizing mentions of entities in text and annotating them with the type they belong to. Typical named-entity extraction systems handle from five to thirty types, recognizing e.g. mentions of people, places, organizations, etc. The particular system we studied, Resporator [Prager, et al, 2000], is unusual in that it supported, at the time of our study, over 90 types of semantic annotations.

The results of this kind of IE can be used for a variety of purposes, such as search, indexing, classification, data- or knowledge-base population, etc. We have been working on the latter, that is, a system that takes the output of an annotation-based IE system and uses the extracted information to populate a knowledge-base. The knowledge-base is represented in a formal logic-based language (OWL), and is built on an ontology of the basic types of entities in the domain of interest. The strength of a formal language like OWL is that a precise model-theoretic semantics [Patel-Schneider, et al, 2003] exists that specifies what information can be inferred from the explicit content of a knowledge-base. This inferred information would otherwise be inaccessible.

One notable characteristic of the ontologies used in knowledge-based system is that they are developed based on observation of the actual universe, *not from observation of language.* We believe this point is critical, as it is the root of a significant integration problem between the IE output and the knowledge-base. This integration problem manifests itself in a variety of ways, including distinguishing classes from instances, mentions from entities, relations from roles, etc. In most cases, the central theme is that the *semantics of the extracted information is ambiguous* with respect to the distinctions made in the ontology, making integration difficult.

In this paper, we present in detail one such ambiguity – the use of *multiple annotations* on a span of text – and show seven different ways that multiple annotations can be formally interpreted. We discuss these seven interpretations, and propose a model by which the semantics can be made more precise.

## Background

To begin with, all the examples in this paper are actual examples except where noted, and any context that makes the meaning of the terms used in the sentences has been preserved. We use in-line XML markup to indicate annotations over spans of text for clarity, however it should be noted that the underlying support infrastructure uses stand-off annotation, which has the advantage of allowing non-nested overlapping spans [Ferrucci and Lally, 2003]. For example, given the sentence:

> Twelve representatives of Turkey arrived in New York today.

If a text analysis engine (TAE) found that "Turkey" is the mention of a country, we will represent that as:

> Twelve representatives of <COUNTRY> Turkey </COUNTRY> arrived in New York today.

We take the intended meaning of an annotation on a span of text to be that the span of text is a *mention* of an *entity* of the *type* named by the annotation.

Multiple annotations occur when a TAE or TAEs assign more than one type to a span of text. For example, one might say that "Turkey" is also an animal, so the sentence above might be annotated as:

> Twelve representatives of <COUNTRY> <ANIMAL> Turkey </ANIMAL > </COUNTRY> arrived in New York today.

At a first pass, the problem of interpreting the meaning of multiple annotations seems fairly trivial. In the example above, it is clear that the TAE is not using enough context to disambiguate the term, "Turkey", and is conveying the different possible types of the entity mentioned. Formally, this would be represented as disjunction. The waters become murky quite quickly, as in the sentence:

> Latin was spoken in <DATE><CITY>ancient Rome</CITY></DATE>.

In this case, the annotator intends *both* types to be assigned to "Ancient Rome". Formally, this would be represented as conjunction. This issue is complicated by the fact that the system providing the annotation may not know which of the cases (disjunction or conjunction) apply; in the latter example, the *correct* interpretation of the annotation is that "ancient Rome" is the city and the time, but the interpretation intended by the creator

| Class | Example | Logical Interpretation |
|---|---|---|
| Ambiguity | | |
| Uncertainty in annotator | Many <LANGUAGE> <NATIONAL> Norwegians </NATIONAL> </LANGUAGE> own snow shovels. | $\exists x$ name($x$, "Norwegian") $\wedge$ [ Language($x$) $\vee$ Nationality($x$) ] $\wedge$ mentionOf(m01,$x$) |
| Known ambiguity | The population of <CITY> <STATE> New York </STATE> </CITY> is decreasing. | $\exists xy$ name($x$,"New York") $\wedge$ City($x$) $\wedge$ name($y$,"New York") $\wedge$ State($y$) $\wedge$ [ mentionOf(m02,$x$) $\oplus$ mentionOf(m02,$y$) ] |
| Deliberate Ambiguity | The duck asked me to put it on his <BODYPART> <FINANCIAL> bill </FINANCIAL> </BODYPART>.* | Same as above with modal quantifier? Maybe conjunction? |
| Type Hierarchy | <PLACE> <CITY> Salt Lake City </CITY> </PLACE> will host the winter Olympics. | $\exists x$ name($x$, "Salt Lake City") $\wedge$ City($x$) $\wedge$ mentionOf(m03,$x$)<br>$\forall y$ City($y$) $\rightarrow$ Place($y$) |
| Conjunction | | |
| Inherence | <COMPOS> <BODYPART> *Hair* </BODYPART> </COMPOS>, is a well-known musical. | $\exists xy$ name($x$, "Hair") $\wedge$ BodyPart($x$) $\wedge$ name($y$, "Hair") $\wedge$ Composition($y$) $\wedge$ mentionOf(m04,$x$) $\wedge$ mentionOf(m04,$y$) |
| Simple conjunction | Latin was spoken in <DATE><CITY>ancient Rome</CITY></DATE>. | $\exists x$ name($x$, "ancient Rome") $\wedge$ City($x$) $\wedge$ Date($x$) $\wedge$ mentionOf(m05,$x$) |
| Nesting | John arrived at <FACILITY> <CITY> Denver </CITY> airport </FACILITY> today. | $\exists xy$ name($x$, "Denver") $\wedge$ City($x$) $\wedge$ name($y$, "Denver airport") $\wedge$ Facility($y$) $\wedge$ mentionOf(m06,$x$) $\wedge$ mentionOf(m07,$y$) $\wedge$ locatedIn($y$,$x$) |

**Table 1.** Classes of multiple annotations.

of the TAE that produced the annotation may simply be that ancient Rome may be a city, or a time, or both.

Resolving this kind of ambiguity is critical if the extracted information is to be used in a knowledge-based system, in which there is a significant difference between the interpretation of disjunction and conjunction. For example, if a system can recognize that the annotations in the former sentence encode disjunction while those in the latter encode conjunction, then that system can correctly infer that ancient Rome is both a city and a time without incorrectly inferring that there exists a country that is a food.

We found at least seven different ways in which multiple annotations are used to convey meaning, summarized in Table 1 and discussed in the next section. If we are to use the results of large-scale information extraction in a knowledge-based system without human

[*] Contrived example.

intervention (or, at least, quite minimal intervention) we must have a consistent interpretation of the intended meaning; clearly one mechanism that conveys more than seven different intended meanings does not have a consistent interpretation.

## Different meanings for double annotations

We have identified four main classes of intended meanings when *two* annotations are present on a span, as shown in Table 1: ambiguity, type hierarchy, conjunction, and nesting. We present some discussion on more than two annotations per span in the next section.

Table 1 also shows several interesting subclasses. This is not intended to be a comprehensive taxonomy of all possible semantics for multiple annotations. Instead our taxonomy is intended to illustrate a variety of circumstances with distinct semantics. Many of the classes and subclasses we describe can be further decomposed into increasingly subtle (but not necessarily unimportant) semantic distinctions. In the logical interpretation, we explicitly refer to instances of *Mention* by unique symbol names starting with *m*. These instances are intended to refer to the specific span of text in the example column that carries the annotation.

The *logical interpretation* column of Table 1 is not intended to show the correct interpretation of the annotations, but rather a formal representation of *what meaning the TAE is trying to convey in the multiple annotation.*

### *Ambiguity*

In our experiments, the most common intent of a double annotation is to express ambiguity, either on the part of the TAE or as a property of the mention; initial analysis shows that roughly 45% of multiple annotations are intended to convey ambiguity.

In the first case shown in Table 1, the double annotation signifies that the TAE is unsure whether the mention of “Norwegians” is a reference to a language or a nationality. In this case, the answer is clear from the passage itself, however the TAE is clearly not considering the wider context, and probably just looking up the string “Norwegian” in a table. Again, the point is not what the correct annotation should be, but what information the TAE is attempting to convey. In this case, the TAE is expressing its own uncertainty. The logical interpretation attempts to capture this: there is a single entity that the mention refers to, and that entity is a language or a nationality.

In the second case, the double annotation is intended to convey the fact that the mention of “New York” in this usage is ambiguous, as it is the name of a state and the name of a city. Once again we are not so concerned with what the correct annotation should be, but rather a case where the TAE actually knows about a possible ambiguity and would like to make a statement that the mention is ambiguous. The logical interpretation captures this in that there are two entities, one a city and one a state, and the mention refers to one or the other but not both (exclusive or).

An interesting refinement of known ambiguity is *deliberate* ambiguity. In these cases, there is still no single interpretation in which both meanings hold, but both interpretations are intended. In some cases, deliberate ambiguity may appear in a political or legal statement in order to create an impression without making a firm commitment. Another common use of deliberate ambiguity is in puns, as the example shows. Dealing with this formally would require a modal operator, as we need to qualify the disjunction itself. We did not actually find any examples of this case, and Resporator does not produce the annotation in the table.

Roughly 80% of multiple annotations we found that were intended to convey ambiguity expressed the uncertainty of Resporator, typically based on a context-free interpretation of some string. For example, Resporator considers “Turkey” to be a country and an animal – clearly in most actual mentions only one interpretation is correct. Furthermore, Resporator has several specific annotation types whose intention is to convey uncertainty, such as UNO (possibly an organization) and UNP (possibly a person). Resporator assumes that any mention annotated with both UNO and UNP is either a person or organization, i.e. that the uncertainty is only in the disjunction. In these cases, the interpretation of multiple annotations depends on the types – when a mention is annotated with UNP and UNO, it is defined to be a disjunction.

## *Type Hierarchy*

The next most common intended meaning for double annotations is to express a type hierarchy, which accounted for roughly another 35%. In the example in Table 1, the double annotation is intended to mean that “Salt Lake City” is both a place and a city. Again, (and this simply cannot be over-stated), the point is not what the correct annotation should or shouldn’t be, the point is what meaning the TAE intends to convey. In this case, the TAE knows that all cities are places, and is simply adding the place annotation based on this knowledge.

It seems fairly clear that multiple annotations are a poor way to specify a type hierarchy, and this information should be conveyed through some other mechanism.

Note that this is different from a TAE indicating alternative interpretations at multiple levels of specificity. For example:

In <DATE> <YEAR> '69 </YEAR> </DATE> I was 21.

We can imagine a TAE intending the DATE and YEAR annotations to be interpreted as “this appears to be a year, but if it isn’t then it is probably some other sort of date.” In this paper we have chosen to ignore issues of relative uncertainty, mainly because this is difficult to convey to a logic-based system, but also because none of the IE systems we have worked with use any notion of certainty for annotations.

## *Conjunction*

While our informal investigations clearly indicated that there were more multiple annotations expressing ambiguity than any other class, most *people* seem, at first glance, to take multiple annotations to signify conjunction, even though this accounted for 20% of actual mentions. Of course one could view type hierarchy annotations as conjunctions – at the sentence level they do express membership in multiple types, however they are also saying something more.

We have broken conjunction into two classes: simple conjunction and *inherence*.

The example in Table 1 for simple conjunction shows a case in which there is a single entity in the world which is both a place and a time (it is a place in time). Again, this is not a question of what is correct, but what the TAE intends to convey.

Inherence is a philosophical term that refers to an essential connection between entities in the world [Hetherington, 1984]. Although we have avoided making ontological statements thus far, in this case we feel it cannot be avoided. What the TAE intends to convey here is that there are *two entities represented by a single mention.* This is a problem that crops up repeatedly in ontology: consider a statue and the marble from which it is made. Many would consider the statue to be a separate entity that inheres to the marble – it cannot exist without it. Consider a book in a library: I can have read the book without ever touching the book. Again we can consider a book to be two entities, an abstract work that inheres to the physical object.

Inherence may also account for lexical annotations. For example:

> The bus took <NOUN> <PERSON> Chris <PERSON> <NOUN> to the airport.

This is not strictly inherence in the philosophical sense, however we do have multiple entities being referred to: the word "Chris", which is a noun, and the person, Chris, who is not a noun. This needs further consideration, however it is not a principle concern since the knowledge-base we are populating will simply ignore lexical annotation.

### *Nested Annotations*

Most cases where annotations contain each other can simply be treated as separate annotations on separate mentions, and the fact that one mention contains the other is not relevant to the semantics. For example:

> Mt. McKinley is located <PLACE> <DISTANCE> 100 miles </DISTANCE> north of Anchorage </PLACE>.

We have found several cases, however, where a TAE uses nesting of annotations to convey relational information. In the example in Table 1, the TAE is expressing that "Denver Airport" is an airport located in Denver. While this case was not very common in our analysis, relation extraction is just beginning to get under way. This is not an appropriate way to carry that kind of semantics.

## Triple Annotations and Beyond

It is clearly possible to have sets of annotations that combine more than one of the above classes, e.g.:

> I visited a friend in <PLACE> <CITY> <STATE> New York </STATE> </CITY> </PLACE>.

Here CITY/STATE are expressing ambiguity while PLACE/CITY and PLACE/STATE are both subtypes. Clearly this possibility creates an infinite number of possible interpretations, as there is no limit on the number of annotations a span of text may have. As a practical matter, we found in Resporator's corpora that fewer than 1% of annotated spans had more than four annotations, but this still results in a large number (on the order of 42) possible interpretations for quadruple annotations (each pair may have one of the seven meanings in Table 1), making automatic mapping to a knowledge-based quite hard.

## Encoding Classes of Multiple Annotations

The basic point here is that we have found TAEs have a need to communicate what they know about a particular span of text, and that multiple annotations have been used to encode seven semantically distinct things. The solution we propose is therefore quite simple: don't use multiple annotations to represent seven different things.

The infrastructure support for text analytics at IBM supports full feature structures as part of text annotations [Ferrucci and Lally, 2003]. It is likely that this multiple annotation ambiguity stemmed from both ignorance of the ambiguity itself and the lack of representational capability in previous systems for more than just a label per annotation.

Our initial proposed solution is to use feature structures to encode the meaning intended by the TAE. It is important to reiterate here that we are not, and have not in the paper, been dealing with situations where the TAE is wrong – we are dealing with the TAE expressing what it knows, whether that is right or wrong.

Given a TAE that knows which of the seven categories it actually means, it can include this information in its output. For example, encoding features as XML attirubtes, we could annotate the first example from Table 1 as:

> Many <LANGUAGE mult=uncertain> <NATIONAL mult=uncertain> Norwegians </NATIONAL> </LANGUAGE> own snow shovels.

This approach is simplistic in that it only handles double annotations. For triple or more, the features would need to refer to the relationship between each *pair* of annotations.

Our work is in the initial stages yet, and we are still studying and documenting the problem. This approach has not been attempted in any TAEs.

## Conclusion

In an effort to automatically populate a knowledge-base from information extraction systems that produce annotations over spans of text, we have found at least seven semantically distinct uses of multiple annotations on a span. The problem we have identified is that the extraction systems need to communicate these seven kinds of things about the entities referred to in text, and have been using the same representation to encode all of them. This creates a problem for semantic integration, since one representation on the IE side maps to one of seven representations on the KB side, and we can not automate the mapping of the intended semantics.

The work is still in the initial stages, and at this point we are studying and documenting the problem. Our proposed solutions, although promising, are preliminary and have not been attempted in any real system.

## References

Ferrucci, David and Adam Lally. 2003. Accelerating Corporate Research in the Development, Application and Deployment of Human Language Technologies. In, *Proceedings of the The Software Engineering and Architecture of Language Technology Systems Workshop (SEALTS).* May 2003, Edmonton, Canada.

Patel-Schneider, Peter, Pat Hayes, and Ian Horrocks. 2003. *The OWL Web Ontology Language Semantics and Abstract Syntax*. W3C Candidate Recommendation. http://www.w3.org/TR/owl-semantics/.

Prager, John, Eric Brown, Anni Coden and Dragomir Radev. 2000. Question-Answering by Predictive Annotation. *Proceedings of SIGIR 2000,* pp. 184-191, Athens, Greece.

Hetherington, S.C. 1984. A Note on Inherence. *Ancient Philosophy* 4: 218-223.