Semantically Rich Human-Aided Machine Annotation

Marjorie McShane, Sergei Nirenburg, Stephen Beale, Thomas O'Hara
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, Maryland, 21250 USA
{marge, sergei, sbeale, tomohara}@umbc.edu

Abstract

This paper describes a semantically rich, human-aided machine annotation system created within the Ontological Semantics (OntoSem) environment using the DEKADE toolset. In contrast to mainstream annotation efforts, this method of annotation provides more information at a lower cost and, for the most part, shifts the maintenance of consistency to the system itself. In addition, each tagging effort not only produces knowledge resources for that corpus, but also leads to improvements in the knowledge environment that will better support subsequent tagging efforts.

1 Introduction

Corpus tagging is a prerequisite for many machine learning methods in NLP but has the drawbacks of high cost, inter-annotator inconsistency and the insufficient treatment of meaning. A tagging approach that strives to ameliorate all of these drawbacks is semantically rich, human-aided machine annotation (HAMA), implemented in the OntoSem (Ontological Semantics) environment using a toolset called DEKADE: the Development, Evaluation, Knowledge Acquisition and Demonstration Environment of OntoSem.

In brief, the OntoSem text analyzer takes as input open text and outputs a text-meaning representation (TMR) that represents its meaning using an ontologically grounded, language-independent metalanguage (see Nirenburg and Raskin 2004). Since the processing leading up to the production of TMR includes, in addition to semantic analysis proper, preprocessing (roughly, segmentation, treatment of named entities and morphology) and syntactic analysis, the overall annotation of text in this approach includes tags relating to all of the above levels. Since the typical input for analysis in our practice is genuine sentences, which are on average 25 words long and contain all manner of complex phenomena, it is not uncommon for the automatically generated TMRs to contain errors. These errors—which can occur at the level of preprocessing, syntactic analysis or semantic analysis—can be corrected manually using the DEKADE environment, yielding "gold standard" output. Making a human the final arbiter in the process means that such long-term complexities as treatment of metaphor, metonymy, PP-attachment, difficult cases of reference resolution and others can be resolved locally while we work on fundamental, implementable automatic solutions.

In this paper we describe the OntoSem/DEKADE environment for the creation of gold standard TMRs, which supports the first ever annotation effort that:

- produces structures that can be used as input for both text generators and general reasoning systems: semantically rich representations of the meaning of text written in a language-independent metalanguage; these representations cover entities, propositions, relations, attributes, speaker attitudes, modalities, polarity, discourse relations, time, reference relations, and more;
- produces semantic tagging of text largely automatically, thus making more realistic and affordable the tagging of large amounts of text in finite time;
- almost fully circumvents the pitfalls of manual tagging, including human tagger errors and inconsistencies;
- produces richer semantic annotations than manual tagging realistically could, since manipulating large and complex static knowl-

Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pages 68–75,
Ann Arbor, June 2005. ©2005 Association for Computational Linguistics

edge sources would be impossible for humans if starting from scratch (i.e., our methodology effectively turns an essay question into a multiple choice one, with most of the correct answers already provided);

- incorporates humans as final arbiters for output of three stages of text analysis (preprocessing, syntactic analysis and semantic analysis), thus maximally leveraging the automated capacity of the system but not requiring of it blanket coverage at this point in its development;
- promises to reduce, over time, the dependence on human input because an important side effect of the operation of the human-assisted machine annotation approach is enhancement of the static knowledge resources – the lexicon and the ontology – underlying the OntoSem analyzer, so that the quality of automatic text analysis will grow as the HAMA system operates, leading to an ever improving quality of raw, unedited TMRs;
- (as a corollary to the previous point) becomes more cost-efficient over time; and
- can be cost-effectively extended to other languages (including less commonly taught languages), with much less work than was required for the first language since many of the necessary resources are language-independent.

Our approach to text analysis is a hybrid of knowledge-based and corpus-based, stochastic methods.

In the remainder of the paper we will briefly describe the lay of the land in text annotation (Section 2), the OntoSem environment (Section 3), the DEKADE environment for creating gold-standard TMRs from automatically generated ones (Section 4), the portability of OntoSem to other languages (Section 5), and the broader implications of this R&D effort (Section 6).

2 The Lay of the Land in Annotation

In addition to the well-known bottlenecks of cost and inconsistency, it is widely assumed that low-level (only syntactic or "light semantic") tagging is either sufficient or inevitable due to the complexity of semantic tagging. Past and ongoing tagging efforts share this point of departure.

Numerous projects have striven to achieve text annotation via a simpler task, like translation, sometimes assuming that one language has already been tagged (e.g., Pianta and Bentivogli 2003, and references therein). But results of such efforts are either of low quality, light semantic depth, or remain to be reported. Of significant interest is the porting of annotations across languages: for example, Yarowsky et al. 2001 present a method for automatic tagging of English and the projection of the tags to other languages; however, these tags do not include semantics.

Post-editing of automatic annotation has been pursued in various projects (e.g., Brants 2000, and Marcus et al. 1993). The latter group did an experiment early on in which they found that "manual tagging took about twice as long as correcting [automated tagging], with about twice the inter-annotator disagreement rate and an error rate that was about 50% higher" (Marcus et al. 1993). This conclusion supports the pursuit of automated tagging methods. The difference between our work and the work in the above projects, however, is that syntax for us is only a step in the progression toward semantics.

Interesting time- and cost-related observations are provided in Brants 2000 with respect to the manual correction of automated POS and syntactic tagging of a German corpus (semantics is not addressed). Although these tasks took approximately 50 seconds per sentence, with sentences averaging 17.5 tokens, the actual cost in time and money puts each sentence at 10 minutes, by the time two taggers carry out the task, their results are compared, difficult issues are resolved, and taggers are trained in the first place. Notably, however, this effort used students as taggers, not professionals. We, by contrast, use professionals to check and correct TMRs and thus reduce to practically zero the training time, the need for multiple annotators (provided the size of a typical annotation task is commensurate with those in current projects), and costly correction of errors.

Among past projects that have addressed semantic annotation are the following:

1. Gildea and Jurafsky (2002) created a stochastic system that labels case roles of predicates with either abstract (e.g., AGENT, THEME) or domain-specific (e.g., MESSAGE, TOPIC) roles. The system

trained on 50,000 words of hand-annotated text (produced by the FrameNet project). When tasked to segment constituents and identify their semantic roles (with fillers being undisambiguated textual strings) the system scored in the 60's in precision and recall. Limitations of the system include its reliance on hand-annotated data, and its reliance on prior knowledge of the predicate frame type (i.e., it lacks the capacity to disambiguate productively). Semantics in this project is limited to case-roles.

2. The goal of the "Interlingual Annotation of Multilingual Text Corpora" project (http://aitc.aitcnet.org/nsf/iamtc/) is to create a syntactic and semantic annotation representation methodology and test it out on six languages (English, Spanish, French, Arabic, Japanese, Korean, and Hindi). The semantic representation, however, is restricted to those aspects of syntax and semantics that developers believe can be consistently handled well by hand annotators for many languages. The current stage of development includes only syntax and light semantics – essentially, thematic roles.

3. In the ACE project (http://www.ldc.upenn.edu/Projects/ACE/intro.html), annotators carry out manual semantic annotation of texts in English, Chinese and Arabic to create training and test data for research task evaluations. The downside of this effort is that the inventory of semantic entities, relations and events is very small and therefore the resulting semantic representations are coarse-grained: e.g., there are only five event types. The project description promises more fine-grained descriptors and relations among events in the future.

4. Another response to the insufficiency of syntax-only tagging is offered by the developers of PropBank, the Penn Treebank semantic extension. Kingsbury et al. 2002 report: "It was agreed that the highest priority, and the most feasible type of semantic annotation, is coreference and predicate argument structure for verbs, participial modifiers and nominalizations", and this is what is included in PropBank.

To summarize, previous tagging efforts that have addressed semantics at all have covered only a relatively small subset of semantic phenomena. OntoSem, by contrast, produces a far richer annotation, carried out largely automatically, within an environment that will improve over time and with use.

3 A Snapshot of OntoSem

OntoSem is a text-processing environment that takes as input unrestricted raw text and carries out preprocessing, morphological analysis, syntactic analysis, and semantic analysis, with the results of semantic analysis represented as formal text-meaning representations (TMRs) that can then be used as the basis for many applications (for details, see, e.g., Nirenburg and Raskin 2004, Beale et al. 2003). Text analysis relies on:

- The OntoSem language-independent ontology, which is written using a metalanguage of description and currently contains around 6,000 concepts, each of which is described by an average of 16 properties.
- An OntoSem lexicon for each language processed, which contains syntactic and semantic zones (linked using variables) as well as calls for procedural semantic routines when necessary. The semantic zone most frequently refers to ontological concepts, either directly or with property-based modifications, but can also describe word meaning extra-ontologically, for example, in terms of modality, aspect, time, etc. The current English lexicon contains approximately 25,000 senses, including most closed-class items and many of the most frequent and polysemous verbs, as targeted by corpus analysis. (An extensive description of the lexicon, formatted as a tutorial, can be found at http://ilit.umbc.edu.)
- An onomasticon, or lexicon of proper names, which contains approximately 350,000 entries.
- A fact repository, which contains real-world facts represented as numbered "remembered instances" of ontological concepts (e.g., SPEECH-ACT-3366 is the 3366th instantiation of the concept SPEECH-ACT in the world model constructed during the processing of some given text(s)).
- The OntoSem syntactic-semantic analyzer, which covers preprocessing, syntactic analysis, semantic analysis, and the creation of TMRs. Instead of using a large, monolithic grammar of a language, which leads to ambiguity and inefficiency, we use a special lexicalized grammar created on the fly for each input sentence (Beale, et. al. 2003). Syntactic rules are generated from the lexicon entries of each of the words in the sentence, and are supplemented by a small inventory of generalized rules. We augment this

basic grammar with transformations triggered by words or features present in the input sentence.
- The TMR language, which is the metalanguage for representing text meaning.

Creating gold standard TMRs involves running text through the OntoSem processors and checking/correcting the output after three stages of analysis: preprocessing, syntactic analysis, and semantic analysis. These outputs can be viewed and edited as text or as visual representations through the DEKADE interface. Although the gold standard TMR itself does not reflect the results of preprocessing or syntactic analysis, the gold standard results of those stages of processing are stored in the system and can be converted into a more traditional annotation format.

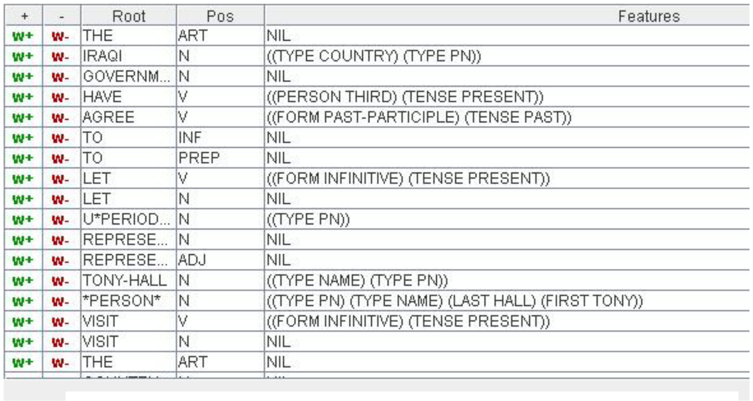
4 TMRs in DEKADE

TMRs represent propositions connected by discourse relations (since space permits only the briefest of descriptions, interested readers are directed to Nirenburg and Raskin 2004, Chapter 6 for details). Propositions are headed by instances of ontological concepts, parameterized for modality, aspect, proposition time, overall TMR time, and style. Each proposition is related to other instantiated concepts using ontologically defined relations (which include case roles and many others) and attributes. Coreference links form an additional layer of linking between instantiated concepts. OntoSem microtheories devoted to modality, aspect, time, style, reference, etc., undergo iterative extensions and improvements in response to system needs as diagnosed during the processing of actual texts.

We use the following sentence to walk through the processes of automatically generating TMRs and viewing/editing those TMRs to create a gold-standard annotated corpus.

> *The Iraqi government has agreed to let U.S. Representative Tony Hall visit the country to assess the humanitarian crisis.*

Preprocessor. The preprocessor identifies the root word, part of speech and morphological features of each word; recognizes sentence boundaries, named entities, dates, times and numbers; and for named entities, determines the ontological type (i.e. HUMAN, PLACE, ORGANIZATION, etc.) of the entity as well as its subparts (e.g., the first, last, and middle names of a person). For the semi-automatic creation of gold standard TMRs, much ambiguity can be removed at small cost by allowing people to correct spurious part-of-speech tags, number and date boundaries, etc., through the DEKADE environment at the preprocessor stage (see Figure 1). Clicking on w+ permits a new POS tag/analysis, and clicking on w-, the more common action, removes spurious analyses. Preprocessor correction is a conceptually simple and logistically fast task that can be carried out by less trained, and therefore less expensive, annotators.



+	-	Root	Pos	Features
w+	w-	THE	ART	NIL
w+	w-	IRAQI	N	((TYPE COUNTRY) (TYPE PN))
w+	w-	GOVERNM...	N	NIL
w+	w-	HAVE	V	((PERSON THIRD) (TENSE PRESENT))
w+	w-	AGREE	V	((FORM PAST-PARTICIPLE) (TENSE PAST))
w+	w-	TO	INF	NIL
w+	w-	TO	PREP	NIL
w+	w-	LET	V	((FORM INFINITIVE) (TENSE PRESENT))
w+	w-	LET	N	NIL
w+	w-	U*PERIOD...	N	((TYPE PN))
w+	w-	REPRESE...	N	NIL
w+	w-	REPRESE...	ADJ	NIL
w+	w-	TONY-HALL	N	((TYPE NAME) (TYPE PN))
w+	w-	*PERSON*	N	((TYPE PN) (TYPE NAME) (LAST HALL) (FIRST TONY))
w+	w-	VISIT	V	((FORM INFINITIVE) (TENSE PRESENT))
w+	w-	VISIT	N	NIL
w+	w-	THE	ART	NIL

Figure 1. Preprocesor Output Editor.

Syntax. Syntax output can be viewed and edited in text or graphic form. The graphic viewer/editor presents the sentence using the traditional metaphor of color-coded labeled arcs. Mouse clicks show the components of arcs, permit arcs to be deleted along with the orphans they would leave, allow for the edges of arcs to be moved, etc. (no graphic of the syntax or semantics browsers/editors are provided due to space constraints).

One common error in syntax output is spurious parses due to contextually incorrect POS or feature analysis. As shown above, this can be fixed from the outset by correcting the preprocessor. However, since the preprocessor will always contain spurious analyses that can usually be removed automatically by the syntactic analyzer, it is not necessarily most time efficient to always start with preprocessor editing. A more difficult, long-term research issue is genuine ambiguity caused, for example, by PP-attachments. While such issues are not likely to be solved computationally in the short term, they can be easily resolved when humans are used as the final arbiters in the creation of gold standard TMRs.

When the correct parse is not included in the syntactic output, either the necessary lexical knowledge is lacking (i.e. there is an unknown word or word sense), or an unknown grammatical construction has been used. While the syntax-editing interface permits spot-correction of the problem by the addition of the necessary arc(s), a more fundamental knowledge-building approach is generally preferred – except when the input is non-standard, in which case systemic modifications are avoided.

Semantics. Within the OntoSem environment, there are two stages of text-meaning representations (TMRs): basic and extended. The basic TMR shows the basic ontological mappings and dependency structure, whereas the extended TMR shows the results of procedural semantics, including reference resolution, reasoning about time relations, etc. The basic and extended stages of TMR creation can be viewed and edited separately within DEKADE.

TMRs can be viewed and edited in text format or graphically. In the latter, concepts are shown as nodes and properties are shown as lines connecting them. A pretty-printed view of the textual extended TMR for our sample sentence, repeated for convenience, is as follows (concept names are in small caps; instance numbers are appended to them).

> *The Iraqi government has agreed to let U.S. Representative Tony Hall visit the country to assess the humanitarian crisis.*

```
AGREE-268
    textpointer   agree
    THEME         MODALITY-200
    AGENT         GOVERNMENTAL-ORGANIZATION-41
    TIME          (< FIND-ANCHOR-TIME)
GOVERNMENTAL-ORGANIZATION-41
    textpointer   government
    RELATION      NATION-56
    AGENT-OF      AGREE-268
NATION-56
    textpointer   Iraq
    RELATION      GOVERNMENTAL-ORGANIZATION-41
MODALITY-200
    textpointer   let
    TYPE          permissive
    SCOPE         TRAVEL-EVENT-272
    VALUE         1
TRAVEL-EVENT-272
    textpointer   visit
    AGENT         SENATOR-447[1]
    DESTINATION   NATION-57
    PURPOSE       EVALUATE-69
    SCOPE-OF      MODALITY-200
SENATOR-447
    textpointer   Representative Tony Hall[2]
    REPRESENTATIVE-OF NATION-40
NATION-40
    textpointer   U.S.
    REPRESENTED-BY SENATOR-447
NATION-57
    textpointer   country
    COREFER       NATION-56
EVALUATE-69
    AGENT         SENATOR-447
    THEME         DISASTER-EVENT-2
DISASTER-EVENT-2
    BENEFICIARY   SET-23
    THEME-OF      EVALUATE-69
SET-23
    MEMBER-TYPE     HUMAN-1342
    BENEFICIARY-OF  DISASTER-EVENT-2
```

Within the graphical browser, clicking on concept names or properties permits them to be deleted, edited, or permits new ones to be added. It also shows the expansion of any concept in text format.

Evaluating and editing the semantic output is the most challenging aspect of creating gold standard TMRs, since creating formal semantic representations is arguably one of the most difficult tasks in all of NLP. If a knowledge engineer determines that some aspect of the semantic representation is incorrect, the problems can be corrected locally or by editing the knowledge resources and rerunning the analyzer. Local corrections are used, for example, in cases of metaphor and metonymy, which we do not record in our knowledge resources (we are working on a microtheory of tropes but it is not yet implemented). In all other cases, resource supplementation is preferred; it can be carried out either immediately or the problem can be fixed locally, in which case a request will be sent to a knowledge acquirer to carry out the necessary resource enhancements.

[1] The concept SENATOR is defined as a member of a legislative assembly.

[2] Collocations of SOCIAL-ROLE + personal name are handled by the preprocessor.

Striking the balance between short-term goals (a gold standard TMR for the given text) and long-term goals (better analysis of any text in the future) is always a challenge. For example, if a text contained the word *grass* in the sense of 'marijuana', and if the lexicon lacked the word 'grass' altogether, we would want to acquire the meaning 'green lawn cover' as well; however, doing this without constraint could mean getting bogged down by knowledge acquisition (as with the dozens of idiomatic uses of 'have') at the expense of actually producing gold-standard TMRs. There are also cases in which a local solution to semantic representation is very easy whereas a fundamental, machine-reproducible solution is very difficult. Consider the case of relative expressions, like *respective* and *respectively*, as used in *Smith and Matthews pleaded innocent and guilty,* ***respectively***. Manually editing a TMR such that the appropriate properties are linked to their heads is quite simple, whereas writing a program for this non-trivial case of reference resolution is not. Thus, in some cases we push through gold standard TMR production while keeping track of – and developing as time permits – the more difficult aspects of text processing that will enhance TMR output in the future.

The gold standard TMR for the sentence discussed at length here was produced with only a few manual corrections: changing two part of speech tags and selecting the correct sense for one word. Work took less than the 10 minutes reported by Brants 2000 for their non-semantic tagging.

5 Porting to Other Languages

Recently the need for tagged corpora for less commonly taught languages has received much attention. While our group is not currently pursuing such languages, it has in the past: TMRs have been automatically generated for languages such as Chinese, Georgian, Arabic and Persian. We take a short tangent to explain how OntoSem/DEKADE can be extended, at relatively low cost, to the annotation of other languages – showing yet another way in which this approach to annotation reaches beyond the results for any given text or corpus.

Whereas it is typical to assume that lexicons are language-specific whereas ontologies are language-independent, most aspects of the semantic structures (*sem-strucs*) of OntoSem lexicon entries are actually language-independent, apart from the linking of specific variables to their counterparts in the syntactic structure. Stated differently, if we consider *sem-strucs* – no matter what lexicon they originate from – to be building blocks of the representation of *word meaning* (as opposed to concept meaning, as is done in the ontology), then we understand why building a large OntoSem lexicon for English holds excellent promise for future porting to other languages: most of the work is already done. This conception of cross-linguistic lexicon development derives in large part from the Principle of Practical Effability (Nirenburg and Raskin 2004), which states that what can be expressed in one language can *somehow* be expressed in all other languages, be it by a word, a phrase, etc. (Of course, it is not necessary that every nuanced meaning be represented in the lexicon of every language and, as such, there will be some differences in the lexical stock of each language: e.g., whereas German has a word for *white horse* which will be listed in its lexicon, English will not have such a lexical entry, the collocation *white horse* being treated compositionally.) We do not intend to trivialize the fact that creating a new lexicon is a lot of work. It is, however, compelling to consider that a new lexicon of the same quality of our OntoSem English one could be created with little more work than would be required to build a typical translation dictionary. In fact, we recently carried out an experiment on porting the English lexicon to Polish and found that a) much of it could be done semi-automatically and b) the manual work for a second language is considerably less than for the first language (for further discussion, see McShane et al. 2004).

To sum up, the OntoSem ontology and the DEKADE environment are equally suited to any language, and the OntoSem English lexicon and analyzer can be configured to new languages with much less work required than for their initial development. In short, semantic-rich tagging through TMR creation could be a realistic option for languages other than English.

6 Discussion

Lack of interannotator agreement presents a significant problem in annotation efforts (see, e.g., Marcus et al. 1993). With the OntoSem semi-automated approach, there is far less possibility of interannotator disagreement since people only correct the output of the analyzer, which is responsible for consistent and correct deployment of the large and complex static resources: if the knowledge bases are held constant, the analyzer will produce the same output every time, ensuring reproducibility of the annotation.

Evaluation of annotation has largely centered upon the demonstration of interannotator agreement, which is at best a partial standard for evaluation. On the one hand, agreement among annotators does not imply the correctness of the annotations: all annotators could be mistaken, particularly as students are most typically recruited for the job. On the other hand, there are cases of genuine ambiguity, in which more than one annotation is equally correct. Such ambiguity is particularly common with certain classes of referring expressions, like *this* and *that*, which can refer to chunks of text ranging from a noun phrase to many paragraphs. Genuine ambiguity in the context of corpus tagging has been investigated by Poesio and Artstein (ms.), among others, who conclude, reasonably, that a system of tags must permit multiple possible correct coreference relations and that it is useful to evaluate coreference based on coreference chains rather than individual entities.

The abovementioned evidence suggests the need for ever more complex evaluation metrics which are costly to develop and deploy. In fact, evaluation of a complex tagging effort will be almost as complex as the core work itself. In our case, TMRs need to be evaluated not only for their correctness with respect to a given state of knowledge resources but also in the abstract. Speed of gold standard TMR creation must also be evaluated, as well as the number of mistakes at each stage of analysis, and the effect that the correction of output at one stage has on the next stage. No methods or standards for such evaluation are readily available since no work of this type has ever been carried out.

In the face of the usual pressures of time and manpower, we have made the programmatic decision not to focus on all types of evaluation but, rather, to concentrate our evaluation metrics on the correctness of the automated output of the system, the extent to which manual correction is needed, and the depth and robustness of our knowledge resources (see Nirenburg et al. 2004 for our first evaluation effort). We do not deny the ultimate desirability of additional aspects of evaluation in the future.

The main source of variation among knowledge engineers within our approach lies not in reviewing/editing annotations as such, but in building the knowledge sources that give rise to them. To take an actual example we encountered: one member of our group described the phrase *weapon of mass destruction* in the lexicon as BIOLOGICAL-WEAPON or CHEMICAL-WEAPON, while another described it as a WEAPON with the potential to kill a very large number of people/animals. While both of these are correct, they focus on different salient aspects of the collocation. Another example of potential differences at the knowledge level has to do with grain size: whereas one knowledge engineer reviewing a TMR might consider the current lexical mapping of *neurosurgeon* to SURGEON perfectly acceptable, another might consider that this grain size is too rough and that, instead, we need a new concept NEUROSURGEON, whose special properties are ontologically defined. Such cases are to be expected especially as we work on new specialized domains which put greater demands on the depth of knowledge encoded about relevant concepts.

There has been some concern that manual editing of automated annotation can introduce bias. Unfortunately, completely circumventing bias in semantic annotation is and will remain impossible since the process involves semantic interpretation, which often differs among individuals from the outset. As such, even agreements among annotators can be questioned by a third (fourth, etc.) party.

At the present stage of development, the TMR together with the static (ontology, lexicons) and dynamic (analyzer) knowledge sources that are used in generating and manipulating it, already provide substantial coverage for a broad variety of semantic phenomena and represent in a compact way practically attainable solutions for most issues that have concerned the computational linguistics and NLP community for over fifty years. Our TMRs have been used as the substrate for question-answering, MT, knowledge extraction, and were also used as the basis for reasoning in the question-answering system AQUA, where they supplied knowledge to enable the operation of the JTP (Fikes et al., 2003) reasoning module.

We are creating a database of TMRs paired with their corresponding sentences that we believe will be a boon to machine learning research. Repeatedly within the ML community, the creation of a high quality dataset (or datasets) for a particular domain has sparked development of applications, such as learning semantic parsers, learning lexical items, learning about the structure of the underlying domain of discourse, and so on. Moreover, as the quality of the raw TMRs increases due to general improvements to the static resources (in part, as side effects of the operation of the HAMA process) and processors (a long-term goal), the net benefit of this approach will only increase, as the production rate of gold-standard TMRs will increase thus lowering the costs.

TMRs are a useful medium for semantic representation in part because they can capture any content in any language, and even content not expressed in natural language. They can, for example, be used for recording the interim and final results of reasoning by intelligent agents. We fully expect that, as the actual coverage in the ontology and the lexicons and the quality of semantic analysis grows, the TMR format will be extended to accommodate these improvements. Such an extension, we believe, will largely involve movement toward a finer grain size of semantic description, which the existing formalism should readily allow. The metalanguage of TMRs is quite transparent, so that the task of converting them into a different representation language (e.g., OWL) should not be daunting.

References

Stephen Beale, Sergei Nirenburg and Marjorie McShane. 2003. Just-in-time grammar. *Proceedings of the 2003 International Multiconference in Computer Science and Computer Engineering*. Las Vegas, Nevada.

Thorsten Brants. 2000. Inter-annotator agreement for a German newspaper corpus. *LREC-2000.* Athens, Greece.

Richard Fikes, Jessica Jenkins and Gleb Frank. 2003. JTP: A system architecture and component library for hybrid reasoning. *Proceedings of the Seventh World Multiconference on Systemics, Cybernetics, and Informatics*. Orlando, Florida, USA.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics* 28:3, 245-288.

Paul Kingsbury, Martha Palmer and Mitch Marcus. 2002. Adding semantic annotation to the Penn TreeBank. (http://www.cis.upenn.edu/~ace/ HLT2002-propbank.pdf.)

Marcus, Mitchell P., Beatrice Santorini and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics* 19.

Marjorie McShane, Margalit Zabludowski, Sergei Nirenburg and Stephen Beale. 2004. OntoSem and SIMPLE: Two multi-lingual world views. *Proceedings of ACL-2004 Workshop on Text Meaning and Interpretation*. Barcelona, Spain.

Sergei Nirenburg, Stephen Beale and Marjorie McShane. 2004. Evaluating the performance of the OntoSem semantic analyzer. *Proceedings of the ACL Workshop on Text Meaning Representation*. Barcelona, Spain.

Sergei Nirenburg and Victor Raskin. 2004. *Ontological Semantics*. The MIT Press.

Emanuele Pianta and Luisa Bentivogli. 2003. Translation as annotation. *Proceedings of the AI*IA 2003 Workshop "Topics and Perspectives of Natural Language Processing in Italy."* Pisa, Italy.

Massimo Poesio and Ron Artstein. 2005. The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. *Proceedings of the ACL 2005 Workshop "Frontiers in Corpus Annotation II, Pie in the Sky".*

David Yarowsky, Grace Ngai and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. *Proceedings of HLT 2001, First International Conference on Human Language Technology Research,* San Diego, California, USA.