# Fully Embedded Type Systems for the Semantic Annotation Layer

# Ekaterina Buyko & Udo Hahn

Jena University Language & Information Engineering (JULIE) Lab Fürstengraben 30, 07743 Jena, Germany {buyko|hahn}@coling-uni-jena.de

## **Abstract**

We extend an integrated linguistic annotation type system which already covers the formal document layer, lexical and morphosyntactic linguistic specifications by a semantic layer for annotation types. The extension is exemplified using the fairly limited set of semantic entities from ACE-style newspaper language domain annotation efforts. We also suggest a solution for those domains where the number of semantic entities is usually several orders of magnitude larger and, hence, much more diverse. As an example, we deal with the life sciences domain where we define a clean interface between generic semantic annotation types and the highly elaborated conceptual structure of bio domain ontologies. Furthermore, we demonstrate by means of comprehensive use cases the benefits of merging various linguistic annotation layers under the umbrella of an integrated type system.

## 1 Introduction

Human language technology development and natural language engineering activities, in the past years, have led to the creation of considerable amounts of linguistic meta data at virtually all linguistic levels – ranging from lexical to morpho-syntactic and semantic specifications. This work has usually focused on linguistic annotations only from a single-level perspective, e.g., POS tags, phrase structure descriptions, or semantic specifications. Recently,

however, the NLP community shifted its attention to combine and merge different kinds of annotations for single or even for multiple annotation layers (cf., e.g., the activities reported by the *Annotation Compatibility Group* (Meyers, 2006)).

From a system engineering perspective, a common specification framework that would connect different levels of linguistic analysis is recognized as crucial for successfully combining and merging linguistic annotations (cf., e.g., Bird and Liberman (2001), Ide et al. (2003), Ferrucci and Lally (2004)).

In this paper, on the one hand, we want to provide specifications for semantic annotations where various proposals can be integrated within a generic framework in a coherent way, in addition to already integrated linguistic annotation layers (e.g, morphology or syntax). On the other hand, given the formal potential of semantic type hierarchies (e.g., feature inheritance), we want to sort out commonalities they share so that the increasing levels of abstractions they implicitly contain can be made explicit. In particular, we propose a clean demarcation line in terms of an interface definition for large-coverage lexicons or ontologies which hook up to generic semantic annotation units. This strategy will allow us to avoid the duplication of information held in different language resources (lexicons, ontologies, etc.) at the more abstract level of semantic meta data.

## 2 Related work

The design of annotation schemata for language resources and their standardization is often limited to single layers of linguistic analysis, e.g., syntactic or semantic analysis. Syntactic annotation schemata,

such as the one from the Penn Treebank (Marcus et al., 1993), or semantic annotations, such as those underlying ACE (Automatic Content Extraction Program) (Doddington et al., 2004) or TIMEML (Pustejovsky et al., 2003a), are increasingly considered as a *de facto* standard.

Recently, however, the NLP community has started to combine different kinds of linguistic annotations. Major bio-medical corpora, such as GE-NIA (Ohta et al., 2002) or PennBioIE, incorporate several layers of linguistic information in terms of morpho-syntactic, syntactic and semantic annotations within one corpus.

Assembling various annotation levels, however, did not result in an annotation scheme for a complete NLP pipeline as needed, e.g., for information extraction or text mining tasks. This lack was mainly due to missing standards for specifying comprehensive NLP software architectures. Proposals of annotation schemata for a complete NLP pipeline usually shared their explicit linkage to a specific NLP tool suite or NLP system and thus suffer from the provision of a generic annotation framework that can be re-used in other developmental activities (Buitelaar et al. (2003), Schäfer (2006)).

Recently started international standardization frameworks such as LAF under the auspices of the ISO TC37 SC WG-1-1 (Ide et al., 2003), and the emergence of generic NLP frameworks such as UIMA (Unstructured Information Management Architecture) (Ferrucci and Lally, 2004)), reflect the community's intention to create such integrated annotation schemata. Several annotation schemata (type systems in UIMA jargon) were developed within the UIMA framework (Hahn et al. (2007a), Piao et al. (2007)). These type systems indeed do cover the results of linguistic pre-processing, as well as document meta and structure analysis, but they still lack more elaborate specifications at the semantic level of annotation (e.g., relations and events). Furthermore, it is not at all clear how these type systems could be extended when already existing semantic annotation schemata or terminologies come into the play.

Welty and Murdock (2006) introduced in their work HUTT (Hierarchical Unified Type Taxonomy),

an UIMA type system integrating a variety of established information extraction taxonomies (e.g, ACE (Doddington et al., 2004), TIMEBANK (Pustejovsky et al., 2003b), etc.).

As an alternative, we use existing generic (*core*) type systems as a backbone for further definitions and extensions aiming, e.g., at information extraction tasks. We integrate established semantic category systems and ontologies from the newswire and biomedical domains, respectively. Furthermore, we show by means of comprehensive use cases the extended type systems in action and demonstrate the benefits of combining various linguistic annotations under the umbrella of a generic type system.

## 3 Graph-based Annotation Models

The uniform representation of annotated data is recognized as crucial for the interoperability between various linguistic (e.g., morpho-syntactic, syntactic, semantic) annotation layers. In particular, graph-based annotation models were shown to be especially appropriate for linguistic annotation. They provide the scaffold for the design of annotation schemata. The implementation of annotation schemata is then realized within annotation frameworks implementing the model.

We first introduce the graph-based annotation model in a more detailed way and then turn to some annotation schemata based on this model.

## 3.1 Annotation Graphs

There are two basic concepts in the graph-based annotation model, *viz.* nodes and edges. *Nodes* represent feature structures providing the annotation content. Nodes are linked by *edges* either to the original subject of analysis or to other feature structures. According to these specifications, an annotation is a graph which represents a feature structure.

Ide and Suderman (2007) formalize the graph-based model as a part of the LAF, a general framework for representing annotations. A graph of annotations G is defined as a set of vertices V(G) and a set of edges E(G). Vertices and edges can be labelled with features. A feature is characterized as a quadruple (G, VE, K, V), where G is a graph, VE is a vertex or edge in G, K is a feature name and V a feature value.

http://bioie.ldc.upenn.edu

Frameworks implementing the graph-based annotation model are, e.g., LAF and UIMA. LAF provides the GRAF (Graph-based Format) (Ide and Suderman, 2007), an XML serialization format of the model, while UIMA comes with CAS (Common Analysis Structure), an object-oriented implementation of the model (Götz and Suhre, 2004). As UIMA provides a platform for the development and deployment of large-scale NLP applications, it is attracting more and more attention in the NLP community. In the following, we will focus on CAS, the data model of the UIMA framework.

# 3.2 UIMA Common Analysis Structure

Common Analysis Structure (CAS) is a part of the system that controls the data flow in the UIMA architecture. CASes contain the original subject of analysis and annotations in the format of CAS objects. An annotation associates one CAS object with a region in the subject of analysis (e.g., the start and the end positions in the document).

Each CAS object (so-called *type*) consists of *feature structures*. Features specify slots within types and can either have primitive values such as integers or strings, or reference other instances of types in a CAS. Types can be extended via the inheritance mechanism. All types in a CAS derive from the basic type UIMA.TCAS.ANNOTATION. and thus inherit the basic annotation features, *viz. begin* and *end* (referencing spans of annotations in the subject of analysis). Accordingly, CASes provide the platform for the design of annotation schemata (*annotation type systems* in the UIMA jargon).

#### 3.3 Annotation Type Systems

The definition of proper annotation schemata (the definition of labels and their interrelations) is usually carried out within an annotation framework implementing the annotation model. Recently some annotation schemata (type systems) were developed within the UIMA Framework.

The generic type system of Hahn et al. (2007a) aims to provide a core domain-independent type system for linguistic annotation which is extensible to domain-specific type systems. We present here the basic concepts of this type system.

**Multi-layered type system** – The type system consists of five annotation layers which cover the an-

notation of analysis results from the main parts of an NLP pipeline.

- the Document Meta layer represents the bibliographical information of the document and the information about the document content;
- the Document Structure & Style layer represents the formal structure of the document, e.g., its division in text body and title, paragraphs and sections, etc.;
- the Morpho-Syntax layer contains types necessary for the annotation of text segmentation (e.g., tokenization), and stores the results of the morpho-syntactic analysis such as lemmatization, POS tagging, etc.;
- the Syntax layer consists of types for the representation of parsing results (shallow and full parsing, dependency- or constituency-based parsing);
- the Semantics layer currently provides only basic types for the annotation of named entities (with pending extensions to relations and events).

Core types – Each layer contains core annotation types that are defined as domain-independent, e.g., POSTag at the Morpho-Syntax layer, Constituent at the Syntax layer, or Entity at the Semantics layer.

Specialized types — The core types can then be extended by more specialized types. POSTag, e.g., might be extended by types that represent specific tag sets, e.g., PennPOSTag (Marcus et al., 1993) or GeniaPOSTag (Ohta et al., 2002). For the semantic types we find potential extensions to the biomedical domain in terms of types such as Gene or Organism.

**Implementation in CAS** – All feature structures are first defined framework-independently in UML (Unified Modelling Language).<sup>2</sup> The type system is implemented as an UIMA CAS type system.

In our work, we employ this type system as a backbone for further extensions. Our extensions affect the *Semantics* layer only. First, we define additional semantic core types needed for information

<sup>&</sup>lt;sup>2</sup>http://www.uml.org/

Figure 1: Semantics Layer in a Multi-layered UIMA Annotation Scheme in UML Representation

extraction applications, i.e., types representing relations and events. We then extend the core semantic types to the domain-specific types, in particular for the newswire and biomedical domain.

# 4 Core Semantics Layer — Fundamental Types

For information extraction tasks we extend the original *Semantics* layer of Hahn et al. (2007a) by introducing new core types, *viz.* Relation, Event, and EventArgument (see Figure 1). These new type definitions are related to already established semantic annotation schemata (cf., e.g., Doddington et al. (2004)).

We distinguish at the Semantics layer between types which refer to instances or states in the real world (e.g., Entity) and types which refer to linguistic data such as text spans that contain mentions of these objects (e.g., EntityMention). The CAS objects of the type EntityMention, RelationMention, EventMention which are coreferent are aggregated by the CAS object of the type Entity, Relation or Event, respectively. The non-mention types extend the uima.cas.TOP type and thus do not dispose of the features begin and end referring to the text span in contrast to the *mention* types which extend the Annotation type. The non-mention types enhance the uima.cas.TOP type with two additional features, viz. mentions and resourceID. The feature mentions refers to all mentions of the entity or relation in the text collection, while the feature resourceID refers to any resource entry which is unique for this entity (e.g., a pointer to an ontology or a gazetteer).

The Mention types enhance the Annotation type with two default features, *specificType* which specifies the type of the mention and usually refers to a terminology, and the feature *ref* referring to the instance of the entity, relation or event, accordingly. This default feature structure is supplied by the additional features specific for entity, relation and event, accordingly. In the following we describe the Mention types in a more detailed way.

EntityMention refers to the mentions of named entities in the text and provides an additional feature, *mentionLevel*, which specifies the mention level, e.g., *pronominal* or *nominal*.

RelationMention stands for a binary semantic relation between entities. The relation is considered as an ordered pair of entities. The type ArgumentMention represents the argument of the relation and refers through the feature *ref* to the particular EntityMention. The feature *role* assigns the role to the entity in the relation under scrutiny. The feature *specificType* describes the relation in more depth, supplying a subtype (usually *is-a*) of the original relation, e.g., locatedIn is a *specificType* of a Part-of relation.

EventMention stands for a special kind of semantic relations between entities (Doddington et al., 2004), namely those representing n-ary semantic relations. The EventMention type is supplied with seven additional features (see Figure 1). The begin and end features refer to the text region that marks the event occurrence (e.g., word). arguments is a feature which contains an array of n ArgumentMention(s). An Argument-Mention refers through the feature mention to the EntityMention in the text as an event participant. The feature role assigns a certain role to the argument in the event under scrutiny. Polarity is needed to distinguish between true and negated events. The features tense and modality characterize the time the event/relation occurs (e.g., future, past) and the information about how certain this event/relation is (e.g., asserted, hypothetical), respectively.

The introduced types constitute a platform ready for domain-specific extensions. In the following, we show how to envelop already established semantic resources within such an annotation type system.

# 5 Domain-Specific Extensions of the Semantics Layer

We extend here the core semantics type system with domain-specific semantics for the newswire and the biomedical domains (see Figure 2). For the

newswire area we selected the ACE taxonomy. In the biomedical domain we cover parts of the areas of immunogenetics and regulation of gene expression.<sup>3</sup>

## 5.1 Newswire Domain

The general objective of the Automatic Content Program (ACE) is to develop information extraction technology, with focus on entity, relation and event recognition (Doddington et al., 2004). For this task, a two-level taxonomy is provided to distinguish between types and subtypes in the annotated newspaper data. The entity taxonomy contains types such as Person, Organization, Location; subtypes of Person are e.g., Group, Individual. The relation taxonomy contains types such as Physical, Part-Whole; subtypes of Physical are e.g., Located, Near. The event taxonomy contains types such as Life, Contact; subtypes of Life are e.g., Be-Born, Marry.

We represent the ACE taxonomy as a CAS type system by creating CAS types from ACE types (see Figure 3). The ACE Semantics type system is an extension of the core type system just introduced. All ACE CAS types extend the core semantic types EntityMention, RelationMention or EventMention, accordingly (e.g., Person(Person) extends EntityMention, Part-Whole(Part-Whole) extends Relation-Mention, while Conflict(Conflict) extends EventMention).

The ACE subtypes can be represented in CAS using the feature *specificType* that describes the semantic type in a more detailed way (e.g., *specificType* of Person may be Individual). In this way, we represent the two-level ACE taxonomy as ACE types using the *specificType* feature.

In order to dock the complete ACE corpus to the UIMA Framework, we converted the ACE XML Document Type Definition (DTD) to the CAS type system definition by creating CAS types from DTD elements independent of the core type system. Every element in the DTD becomes a type in the CAS. The features of the types are the children and attributes of the corresponding element in the DTD. The complete ACE annotation was converted this way to the CAS representation.

Both type systems are connected by a mapping file which allows for the conversion of the CAS ACE annotation based on the ACE type system to the ACE *Semantics*. The benefit of the creation of the ACE *Semantics* type system is the compatibility to the core type system. The ACE semantic types extend the core semantic types and can be analyzed by tools without any specifications for the original ACE annotation schemata (i.e., the ACE DTD).

#### 5.2 Biomedical Domain

In the biomedical domain available ontological resources play a considerable role in the establishment of the annotation terminology.<sup>4</sup> When we move to the automatic annotation of biomedical documents within an NLP framework, we have to cope with the interrelations between the terminology as available in an ontology and the type system for semantic annotation.

The ontology languages being currently used (e.g., OWL (Patel-Schneider et al., 2002)) are richer in their expressivity (e.g., multiple inheritance) than a graph-based annotation model implemented in CAS. Therefore, the aim is certainly not to reimplement ontologies as CAS type systems but rather to provide a linkage of CAS annotations to the original ontological resources via a clean interface definition.

For the biological application, we thus designed a type system which covers a substantial portion of e.g.,, the area of the *immunogenetics*. As ontological resources we used, e.g., Gene Ontology, IMR Ontology and Cell Ontology.<sup>5</sup>

CAS annotations are linked to the original ontological source by the feature *specificType* that provides the ontology source and entry identifier. The information about this mapping between CAS types

<sup>&</sup>lt;sup>3</sup>The field of *immunogenetics* is the domain focus in the STEMNET project (http://www.stemnet.de). The field of *regulation of gene expression* is one of the domain focus areas in the BOOTSTREP project (http://www.bootstrep.org).

<sup>4</sup>http://obo.sourceforge.net

<sup>&</sup>lt;sup>5</sup>http://obofoundry.org

Figure 3: Extensions of the Semantics Layer for the ACE Semantics

and their ontological entries is indexed in mapping files which can be used by NLP components requiring the information about the ontological representation of CAS types.

In the following we demonstrate the extended type system in action within the UIMA framework.

## 6 Use Cases

The UIMA framework not only provides a formal specification layer based on UML,<sup>6</sup> but also comes with a run-time environment for the interpretation of the type system specifications. This turned out to be useful for implementing systems based on UIMA specifications some of which we introduce below.

## 6.1 NLP Pipelines

For the purposes of information extraction and retrieval in the area of immunogenetics, we set up the STEMNET pipeline (Hahn et al., 2007b) which contains a variety of syntactic and semantic components and a database indexer. For the pipeline configuration we took the core type system and the immunogenetics semantics type system. We represent all results of the NLP processing in the CAS format and maintain the linkage to the original source ontologies. This allows for example for the indexing of annotation results with ontological entries for information retrieval applications. A further benefit would be the conversion of CAS annotation results into a more expressive representation format (RDF<sup>7</sup> graphs) in order to perform reasoning over results as shown by Welty and Murdock (2006).

## **6.2** Support for supervised approaches

Previous research provides ample evidence that (ML-based) supervised approaches are very effective for named entity recognition and relation extraction (e.g., Settles (2004) or Zhang et al. (2006)). In order to support the generation of training data for supervised approaches, semantically annotated corpora (e.g., ACE) have been built. Usually, semantically annotated corpora lack additional annotation

layers. As the supervised approaches in semantic analysis profit from the linguistic pre-processing annotation (i.e., segmentation, morpho-syntactic and syntactic annotation), corpora should be automatically extended relative to annotation layers not already accounted for.

Since the additional annotations are usually not provided in the format of the original corpora, one has to switch between various formats in order to retrieve the annotations of interest. This decreases the performance of the analysis and can be error-prone.

Because we extended a core type system with domain-specific ACE semantics, we can conveniently represent all results of linguistic analysis together with the ACE semantics as combined in the CAS. The ACE pipeline (see Figure 4) reveals how to deal with various linguistic annotations for the purposes of supervised approaches. The pipeline provides an ACE trainer, an ACE classifier and an ACE evaluator. It starts with data segmentation in the training and test data. Data processing starts with the ACE READER which converts the original ACE data to the CAS representation format. Next, the data is processed by linguistic analysis engines (i.e, segmentation, POS tagging, chunking, parsing). After the data is pre-processed, the training CAS data is sent to the trainer which can extract features from the CAS and produces a classification model. The model is then used by the classifier that processes the CAS test data and writes the results to the CAS. The evaluator compares the results of the classifier with the gold standard.

## 6.3 Interoperability and Visualization

All CAS annotations may be serialized in the XMI format within the UIMA framework. XMI, the XML Metada Interchange format (stand-off annotation), is an OMG<sup>8</sup> standard for the XML representation of object graphs. The CAS annotations can be visualized within the XCAS Visualizer in the UIMA framework. The visualization provides a user-friendly look inside the data.

<sup>6</sup>http://www.uml.org

http://www.w3.org/RDF/

<sup>8</sup>http://www.omg.org

## 7 Conclusion and Future work

In this paper, we proposed an integration of various semantic annotation languages, for general newspaper language as well as for (parts of) the sublanguage of the life science, into the coherent framework of an annotation type system. While previous efforts at lower levels of linguistic analysis, up until the syntactic level, are characterized by a significant degree of consensus in what concerns the basic vocabulary of such specification layers, the semantic layer is much harder to deal with. In essence, the semantics layer might contain a multitude of entities similar to entries in a lexicon or concepts in an ontology. So care has to be taken not to duplicate information contained in these resources at the semantic annotation layer.

Here, type hierarchies might be of help for making higher levels of abstraction explicit. Still, they do not provide any heuristic guidance at which level of specificity one has to stop in terms of increasing the level of detail for semantic annotation.

We here defined a clean interface between generic semantic annotation types and the highly elaborated granularity of the conceptual structure of ontologies (or lexical semantic specifications kept in a lexicon). This way, we avoid the redundant duplication of lexical and conceptual knowledge and keep the number of linguistically relevant semantic entities used for semantic annotation at a manageable size.

**Acknowledgments.** This research was funded by the EC's 6th Framework Programme (4th call) within the BOOTStrep project under grant FP6-028099, and by the German Ministry of Education and Research (BMBF) via its e-Science initiative within the StemNet project (funding code: 01DS001A to 1C).

## References

- Steven Bird and Mark Liberman. 2001. A formal framework for linguistic annotation. Speech Communication, 33(1/2):23–60.
- Paul Buitelaar, Thierry Declerck, Bogdan Sacaleanu, Špela Vintar, Diana Raileanu, and Claudia Crispi. 2003. A multilayered, XML-based approach to the integration of linguistic and semantic annotations. In *Proceedings of EACL 2003 Workshop on Language Technology and the Semantic Web (NLPXML-03)*. Budapest, Hungary, April, 2003.

- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) Program: Tasks, data, & evaluation. In *LREC 2004 – Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 837–840. Lisbon, Portugal, 26-28 May 2004.
- David Ferrucci and Adam Lally. 2004. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.
- Thilo Götz and Oliver Suhre. 2004. Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal*, 43(3):476–489.
- Udo Hahn, Ekaterina Buyko, Katrin Tomanek, Scott Piao, John McNaught, Yoshimasa Tsuruoka, and Sophia Ananiadou. 2007a. An annotation type system for a data-driven NLP pipeline. In *The LAW at ACL 2007 Proceedings of the Linguistic Annotation Workshop*, pages 33–40. Prague, Czech Republic, June 28-29, 2007.
- Udo Hahn, Joachim Wermter, David S. DeLuca, Peter Horn, Rainer Blasczyk, Michael Poprat, and Asad Bajwa. 2007b. STEMNET: An evolving service for knowledge networking in the life sciences. In *Proceedings of the German e-Science* Conference 2007. Baden-Baden, Germany, May 2-4, 2007.
- Nancy Ide and Keith Suderman. 2007. GRAF: A graph-based format for linguistic annotations. In *The LAW at ACL 2007 Proceedings of the Linguistic Annotation Workshop*, pages 1–8. Prague, Czech Republic, June 28-29, 2007.
- Nancy Ide, Laurent Romary, and Eric de la Clergerie. 2003. International standard for a linguistic annotation framework. In *Proceedings of the HLT-NAACL 2003 Workshop on 'Software Engineering and Architecture of Language Technology Systems'*, pages 25–30. Edmonton, Canada, May 31, 2003.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The PENN TREEBANK. Computational Linguistics, 19(2):313–330.
- Adam Meyers. 2006. Annotation compatibility working group report. *Proceedings of the COLING-ACL 2006 Workshop on 'Frontiers in Linguistically Annotated Corpora 2006'*, pages 38–53. Sydney, Australia, 22 July 2006.
- Tomoko Ohta, Y. Tateisi, and J. Kim. 2002. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. HLT 2002 – Proceedings of 2nd International Conference on Human Language Technology Research, pages 82–86. San Diego, CA, USA, March 24-27, 2002.
- Peter F. Patel-Schneider, Pat Hayes, Ian Horrocks, and Frank van Harmelen. 2002. OWL Web Ontology Language; Semantics and Abstract Syntax. [http://www.w3.org/TR/owl-semantics/].
- Scott Piao, Ekaterina Buyko, Yoshimasa Tsuruoka, Katrin Tomanek, Jin-Dong Kim, John McNaught, Udo Hahn, Jian Su, and Sophia Ananiadou. 2007. Bootstrep annotation scheme: Encoding information for text mining. In *Corpus Linguistics* 2007 *Proceedings of the 4th Corpus Linguistics Conference*. Birmingham, England, U.K., July 27-30, 2007.

- James Pustejovsky, José Castaño, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003a. TIMEML: Robust specification of event and temporal expressions in text. In Mark Maybury, editor, New Directions in Question Answering, pages 28–34.
- James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. 2003b. The TIMEBANK corpus. Proceedings of the Corpus Linguistics 2003 Conference, pages 647–656. Lancaster, U.K., 28-31 March 2003.
- Ulrich Schäfer. 2006. Middleware for creating and combining multi-dimensional NLP markup. In *Proceedings of the 5th EACL-2006 Workshop on NLP and XML (NLPXML-2006): Multi-Dimensional Markup in Natural Language Processing*, pages 81–84. Trento, Italy, April 4, 2006.
- Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. *JNLPBA Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 107–110. Geneva, Switzerland, August 28-29, 2004.
- Christopher A. Welty and J.William Murdock. 2006. Towards knowledge acquisition from information extraction. *The Semantic Web ISWC 2006. Proceedings of the 5th International Semantic Web Conference*, pages 709–722. Athens, GA, USA, November 5-9, 2006.
- Min Zhang, Jie Zhang, Jian Su, and GouDong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In COLING'ACL 2006 Proceedings of the Joint 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 825–832. Sydney, Australia, July 17-21, 2006.