Combining Independent Syntactic and Semantic Annotation Schemes

Marc Verhagen, Amber Stubbs and James Pustejovsky
Computer Science Department
Brandeis University, Waltham, USA
{marc,astubbs,jamesp}@cs.brandeis.edu

Abstract

We present MAIS, a UIMA-based environment for combining information from various annotated resources. Each resource contains one mode of linguistic annotation and remains independent from the other resources. Interactions between annotations are defined based on use cases.

1 Introduction

MAIS is designed to allow easy access to a set of linguistic annotations. It embodies a methodology to define interactions between separate annotation schemes where each interaction is based on a use case. With MAIS, we adopt the following requirements for the interoperability of syntactic and semantic annotations:

1. Each annotation scheme has its own philosophy and is independent from the other annotations. Simple and generally available interfaces provide access to the content of each annotation scheme.

2. Interactions between annotations are not defined a priori, but based on use cases.

3. Simple tree-based and one-directional merging of annotations is useful for visualization of overlap between schemes.

The annotation schemes currently embedded in MAIS are the Proposition Bank (Palmer et al., 2005), NomBank (Meyers et al., 2004) and TimeBank (Pustejovsky et al., 2003). Other linguistics annotation schemes like the opinion annotation (Wiebe et al., 2005), named entity annotation, and discourse annotation (Miltsakaki et al., 2004) will be added in the future.

In the next section, we elaborate on the first two requirements mentioned above and present the MAIS methodology to achieve interoperability of annotations. In section 3, we present the XBank Browser, a unified browser that allows researchers to inspect overlap between annotation schemes.

2 Interoperability of Annotations

Our goal is not to define a static merger of all annotation schemes. Rather, we avoid defining a potentially complex interlingua and instead focus on how information from different sources can be combined pragmatically. A high-level schematic representation of the system architecture is given in figure 1.

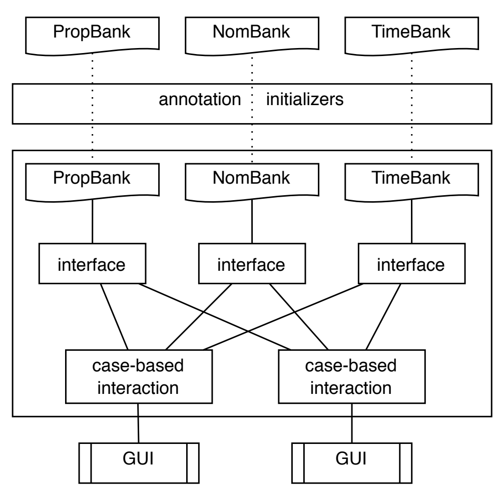

Figure 1: Architecture of MAIS

Proceedings of the Linguistic Annotation Workshop, pages 109–112,
Prague, June 2007.

The simple and extensible interoperability of MAIS can be put in place using three components: a unified environment that stores the annotations and implements some common functionality, a set of annotation interfaces, and a set of case-based interactions.

2.1 Unified Environment

All annotations are embedded as stand-off annotations in a unified environment in which each annotation has its own namespace. This unified environment takes care of some basic functionality. For example, given a tag from one annotation scheme, there is a method that returns tags from other annotation schemes that have the same text extent or tags that have an overlap in text extent. The unified environment chosen for MAIS is UIMA, the open platform for unstructured information analysis created by IBM.[1]

UIMA implements a common data representation named CAS (Common Analysis Structure) that provides read and write access to the documents being analyzed. Existing annotations can be imported into a CAS using CAS Initializers. UIMA also provides a framework for Analysis Engines: modules that can read from and write to a CAS and that can be combined into a complex work flow.

2.2 Annotation Interfaces

In the unified environment, the individual annotations are independent from each other and they are considered immutable. Each annotation defines an interface through which salient details of the annotations can be retrieved. For example, annotation schemes that encodes predicate-argument structure, that is, PropBank and NomBank, define methods like

```
args-of-relation(pred)
arg-of-relation(pred, arg)
relation-of-argument(arg)
```

Similarly, the interface for TimeBank includes methods like

rel-between($event_i$, $event_j$)
events-before(event)
event-anchorings(event)

[1] `http://www.research.ibm.com/UIMA/`

The arguments to these methods are not strings but text positions, where each text position contains an offset and a document identifier. Return values are also text positions. All interfaces are required to include a method that returns the tuples that match a given string:

```
get-locations(string, type)
```

This method returns a set of text positions. Each text position points to a location where the input string occurs as being of the given type. For TimeBank, the type could be `event` or `time`, for PropBank and NomBank, more appropriate values are `rel` or `arg0`.

2.3 Case-based Interactions

Most of the integration work occurs in the interaction components. Specific interactions can be built using the unified environment and the specified interfaces of each annotation scheme.

Take for example, the use case of an entity chronicle (Pustejovsky and Verhagen, 2007). An entity chronicle follows an entity through time, displaying what events an entity was engaged in, how these events are anchored to time expressions, and how the events are ordered relative to each other. Such an application depends on three kinds of information: identification of named entities, predicate-argument structure, and temporal relations. Each of these derive from a separate annotation scheme. A use case can be built using the interfaces for each annotation:

- the named entity annotation returns the text extents of the named entity, using the general method `get-locations(string, type)`
- the predicate-argument annotation (accessed through the PropBank and NomBank interfaces) returns the predicates that go with a named-entity argument, repeatedly using the method `relation-of-argument(arg)`
- finally, the temporal annotation returns the temporal relations between all those predicates, calling rel-between($event_i$, $event_j$) on all pairs of predicates

Note that named entity annotation is not integrated into the current system. As a stopgap measure we use a pre-compiled list of named entities and feed elements of this list into the PropBank and NomBank interfaces, asking for those text positions where the entity is expressed as an argument. This shows the utility of a general method like `get-locations(string, type)`.

Each case-based interaction is implemented using one or more UIMA analysis engines. It should be noted that the analysis engines used for the entity chronicler do not add data to the common data representation. This is not a principled choice: if adding new data to the CAS is useful then it can be part of the case-based interaction, but these added data are not integrated into existing annotations, rather, they are added as a separate secondary resource.[2]

The point of this approach is that applications can be built pragmatically, using only those resources that are needed. It does not depend on fully merged syntactic and semantic representations. The entity chronicle, for example, does not require discourse annotation, opinion annotation or any other resource except for the three discussed before. An a priori requirement to have a unified representation introduces complexities that go beyond what's needed for individual applications.

This is not to say that a unified representation is not useful on its own, there is obvious theoretical interest in thoroughly exploring how annotations relate to each other. But we feel that the unified representation is not needed for most, if not all, practical applications.

3 The XBank Browser

The unified browser, named the XBank Browser, is intended as a convenience for researchers. It shows the overlap between different annotations. Annotations from different schemes are merged into one XML representation and a set of cascading style sheets is used to display the information.

[2] In fact, for the entity chronicle it would be useful to have extra data available. The current implementation uses what's provided by the basic resources plus a few heuristics to superficially merge data from separate documents. But a more informative chronicle along the lines of (Pustejovsky and Verhagen, 2007) would require more temporal links than available in TimeBank. These can be pre-compiled and added using a dedicated analysis engine.

The XBank Browser does not adhere to the MAIS philosophy that all resources are independent. Instead, it designates one syntactic annotation to provide the basic shape of the XML tree and requires tags from other annotations to find landing spots in the basic tree.

The Penn Treebank annotation (Marcus et al., 1993) was chosen to be the first among equals: it is the starting point for the merger and data from other annotations are attached at tree nodes. Currently, only one heuristic is used to merge in data from other sources: go up the tree to find a Treebank constituent that contains the entire extent of the tag that is merged in, then select the head of this constituent. A more sophisticated approach would consist of two steps:

- first try to find an exact match of the imported tag with a Treebank constituent,
- if that fails, find the constituent that contains the entire tag that is merged in, and select this constituent

In the latter case, there can be an option to select the head rather than the whole constituent. In any case, the attached node will be marked if its original extent does not line up with the extent at the tree node.

It should be noted that this merging is one-directional since no attempt is made to change the shape of the tree defined by the Treebank annotation.

The unified browser currently displays markups from the Proposition Bank, NomBank, TimeBank and the Discourse Treebank. Tags from individual schemes can be hidden as desired. The main problem with the XBank Browser is that there is only a limited amount of visual clues that can be used to distinguish individual components from each other and cognitive overload restricts how many annotation schemes can be viewed at the same time. Nevertheless, the browser does show how a limited number of annotation schemes relate to each other.

All functionality of the browser can be accessed at `http://timeml.org/ula/`. An idea of what it looks like can be gleaned from the screenshot displayed in figure 2. In this figure, boxes represent relations from PropBank or NomBank and shaded

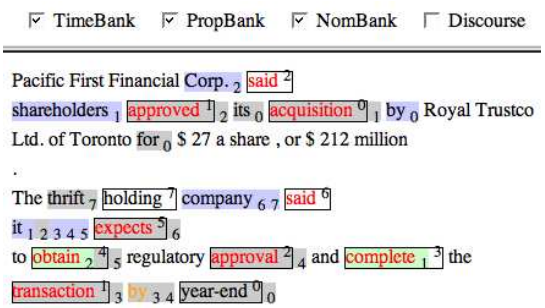

Figure 2: A glimpse of the XBank Browser

backgrounds represent arguments. Superscripts are indexes that identify relations, subscripts identify what relation an argument belongs to. Red fonts indicate events from TimeBank. Note that the real browser is barely done justice by this picture because the browser's use of color is not visible.

4 Conclusion

We described MAIS, an environment that implements interoperability between syntactic and semantic annotation schemes. The kind of interoperability proposed herein does not require an elaborate representational structure that allows the interaction. Rather, it relies on independent annotation schemes with interfaces to the outside world that interact given a specific use case. The more annotations there are, the more interactions can be defined. The complexity of the methodology is not bound by the number of annotation schemes integrated but by the complexity of the use cases.

5 Acknowledgments

The work reported in this paper was performed as part of the project "Towards a Comprehensive Linguistic Annotation of Language", and supported under award CNS-0551615 of the National Science Foundation.

References

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treeb. *Computational Linguistics*, 19(2):313–330.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004. The penn discourse treebank. In *Proceedings of the Language Resources and Evaluation Conference*, Lisbon, Portugal.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

James Pustejovsky and Marc Verhagen. 2007. Constructing event-based entity chronicles. In *Proceedings of the IWCS-7*, Tilburg, The Netherlands.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003. The timebank corpus. In *Proceedings of Corpus Linguistics*, pages 647–656.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.