Integrating Event Frame Annotation into the Open Ontology Forge Annotation Tool

Tuangthong Wattarujeekrit and Nigel Collier

National Institute of Informatics, National Center of Sciences,
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
tuangthong@grad.nii.ac.jp, collier@nii.ac.jp

Abstract. In this paper, we propose a scheme for event frame annotation integrated into the Open Ontology Forge (OOF) annotation tool. This is a key requirement for realization of knowledge description on the Semantic Web. Semantic information contained in each event frame is a set of relationships between a predicate and its arguments. As our aim is to keep OOF flexible for various types of annotation projects, the scheme proposed in this paper is designed based on the specialization three popular schemes: MUC-7's template relation, PropBank's predicate-argument structure and FrameNet's semantic frame.

1 Introduction

This paper provides the scheme for the annotation of event frames which define relationship information between objects or entities and their predicates indicating the event. This scheme is being integrated into Open Ontology Forge (OOF)[1], a free annotation tool created in the PIA project [4].

As the Web of information readable by machines is the central concept of the Semantic Web [2], Web pages require annotation to make instances of objects and events explicit and to show the linkage to the context in which they occur. Thus, the development of annotation tools becomes a focus of the research community (e.g. GATE [6], MnM [9], OntoMat [10]). Like other semantic annotation tools, OOF tries to reduce the effort required to create semantic annotated texts and it focuses mainly on content annotation for Information Extraction (IE) as such we consider issues of large-scale knowledge mark-up, inter-annotator agreement, ease of use by non-linguistics, etc. One of the significant characteristics of OOF is that it not only supports annotation but also provides for the creation of ontology and the linkage between each instance and its occurrence in the text. To provide an environment that integrates annotated texts with ontology promotes knowledge sharing.

The basic aim within the PIA project is to create an automatic information extraction system by applying machine learning to annotated corpora [3]. At present OOF can be used to construct annotated named entities (NEs) and coreference relations [7]. It still lacks though the scheme to support the higher level IE task such

[1] http://research.nii.ac.jp/~collier/resources/OOF/index.htm

as event extraction task which provides facts in terms of relationships between entities obtained from NE and Coreference task. Therefore, the event frame annotation scheme needs to be integrated into OOF. With respect to our interest in the application of IE to special domains such as molecular biology event extraction, we currently plan to annotate molecular biology documents with semantics in terms of event frame style following an extensible version of PropBank's predicate-argument structure [8] description.[2] However, we also take into account other two popular event frame styles among IE research groups (i.e. MUC-7's template relation [5] and FrameNet's semantic frame [1]) in forming the scheme proposed in this paper. We believe that the scheme which incorporates the key features of these three projects will provide OOF the flexibility to be used by other research groups.

2 The Event Frame Annotation Scheme

Annotation of event frames will give web pages some machine readable information describing a set of relationships between entities existing in each proposition. For example, if the event *buying* in the expression "John is buying flowers for Marry." is annotated in an appropriate way, then not only can a machine understand that "John" and "Mary" are persons, but also "John" plays role as "buyer", "flowers" plays role as "bought object" and "Mary" plays role as "receiver" in this *buying* event. So, various applications such as IE can extract these important facts for users. Similarly, the event frame annotation is capable of representing molecular events such as protein-protein interaction also.

In general, different projects have their own perspective on how to define their event frame or how a set of relationships in an event should be represented. Some special distinctions[3] of the event frame's descriptions in three main projects of our focus are illustrated in Fig. 1. However, all styles can be thought of as the general frame-like styles which a relation or a set of relations is specified in an event frame by a related predicate or set of predicates and related arguments or entities.

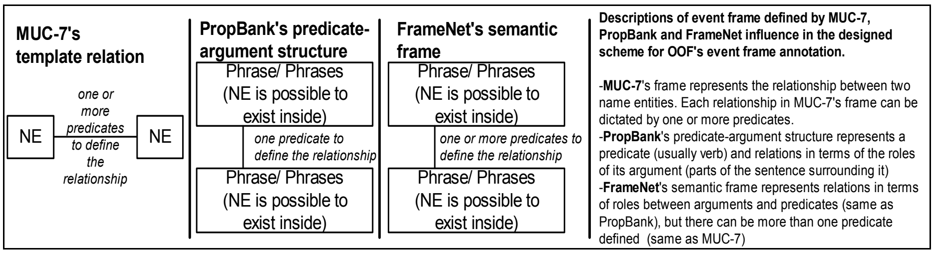

Fig. 1. The abstract view of an event frame from different projects

[2] Reasons why we choose PropBank's event frame style are out of scope of this paper. They will be reported elsewhere.

[3] Due to space limitation, only particular scheme's points will be explained.

2.1 Knowledge Model Issues

The knowledge model of OOF has several similarities to other ontology editors such as Protégé-2000. An OOF ontology is centred around a frame-based knowledge model consisting of classes, properties (slots) and annotations. Classes are related through subsumption in a simple taxonomy. An event frame is managed as a subclass of a root class, called an Event class. The argument participated in an event are represented in the form of Event class's property. Basically, property slot in OOF is a binary relation between a domain (a class) and a range (a value data type). As an event's argument require being filled by more than one value, property of Event class is necessary to be managed as a class rather than just a binary relation. A main predicate of an event is modelled on a property of Event class as well. Moreover, instances of basic class types in OOF are not abstract concept, but a surface-level representation of the concept appearing in the document, in the form of texts or images. Contrary to other class types, event instances are abstract entities but the predicate itself and the arguments are realized as annotations in the text.

2.2 User Interface

As shown in the planned design of the new version of OOF in Fig. 2, OOF provides a capability to view ontology, a Web page and annotated information concurrently.

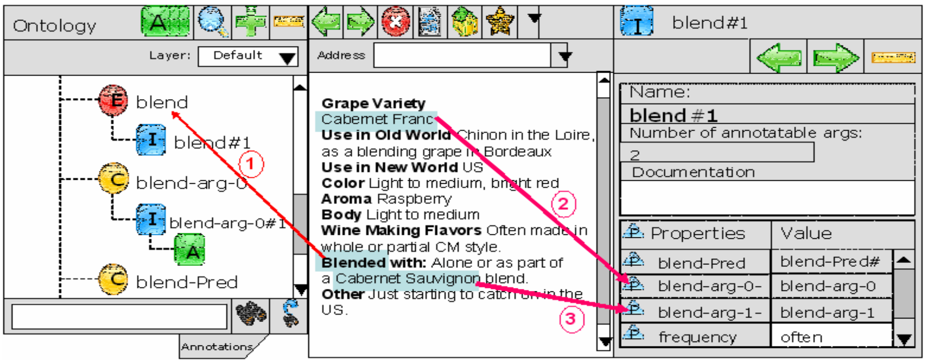

Fig. 2. Event annotation screen shots

For event annotation, a user has to create a particular Event class first (cf. the red E-icon). This process leads to the automatically construction of classes for the predicate and arguments which are defined as properties of the Event class. Then, a user can start annotating an event existing in a text by dragging and dropping the main predicate to the Event class for that event. As shown in Fig. 2, the text "Blended" is highlighted and assigned to the Event class *blend* (cf. arrow #1) to create an instance *blend#1* representing the event. Next, instances for event's arguments can be captured by highlighting some text elements and then using a hot key combination. In this example, the text "Cabernet Franc" and "Cabernet Sauvignon" are captured for filling argument slots *blend-arg-0* and *blend-arg-1*, respectively (cf. arrow #2 and arrow #3).

3 Discussion

There have been several annotation tools such as GATE of which rather focusing on the annotation process embedded with language processing tools (e.g. POS tagger, tokeniser) than the ontology editing; MnM of which distinct property is the supporting of various representation language (e.g. DAML+OIL, RDF, WebOnto); and OntoMat which provides many of the same features as OOF including ontology editing. In contrast to these tools, the focus of our design is highlighting the role of the predicate occurred in the text as the centre of the occurring event. The event itself is represented as an individual object class rather than represented as a property of participated entity. We believe that our thinking of predicate which is much closer to linguistic perspective would allow OOF to be flexible for various event annotation styles.

The OOF has progressed forward in concerning more flexible scheme for event annotation. However, OOF still requires the extension of event annotation scheme in order to support nontrivial aspects such as to represent sequences of events.

4 Conclusion

We briefly presented the main scheme for semantic annotation of event frame being integrated in Open Ontology Forge (OOF) tool, with the design to cover various styles of event frames. The capability both to create ontology and to annotate texts as well as to provide the linkage from the ontology instance to where it exists in texts makes the OOF annotation tool worthwhile for Semantic Web applications. Current version is downloadable from http://research.nii.ac.jp/~collier/OOF/index.htm. We plan on releasing new version included event frame annotation capability in January 2005.

References

1. Baker, C., Fillmore, C., and Lowe, J.: The Berkeley FrameNet project. In *Proc. of COLING-ACL*, Montreal (1998)
2. Berners-Lee, T., Fischetti, M., and Dertouzos, M.: Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web. Harper, San Francisco, September (1999)
3. Collier, N., and Takeuchi, K.: PIA-Core: Semantic Annotation through Example-based Learning. In *Proc. of the 3rd LREC*, Las Palmas, Spain (2002)
4. Collier, N., Takeuchi, K., Kawazoe, A., Mullen, T., and Wattarujeekrit, T.: A Framework for Integrating Deep and Shallow Semantic Structures in Text Mining. In *Proc. of the 7th KES*, UK (2003)
5. DARPA. In *Proc. of MUC-7*, Fairfax, VA, USA (1998)
6. GATE: http://gate.ac.uk/
7. Kawazoe, A. and Collier, N.: An Ontologically-motivated Annotation Scheme for Coreference. In *Proc. of SWFAT*, Nara, Japan (2003)
8. Kingsbury, P. and Palmer, M.: From Treebank to PropBank. In *Proc. of the 3rd LREC*, Las Palmas (2002)
9. MnM: http://kmi.open.ac.uk/projects/akt/MnM/
10. OntoMat: http://annotation.semanticweb.org/ontomat/