CAT: the CELCT Annotation Tool

Valentina Bartalesi Lenzi, Giovanni Moretti, Rachele Sprugnoli
CELCT
via Sommarive 18, Povo (TN), Italy
valentina.bartalesi@isti.cnr.it, moretti@celct.it, sprugnoli@celct.it

Abstract

This paper presents CAT - CELCT Annotation Tool, a new general-purpose web-based tool for text annotation developed by CELCT (Center for the Evaluation of Language and Communication Technologies). The aim of CAT is to make text annotation an intuitive, easy and fast process. In particular, CAT was created to support human annotators in performing linguistic and semantic text annotation and was designed to improve productivity and reduce time spent on this task. Manual text annotation is, in fact, a time-consuming activity, and conflicts may arise with the strict deadlines annotation projects are frequently subject to. Thanks to its adaptability and user-friendly interface, CAT can positively contribute to improve time management in annotation project. Further, the tool has a number of features which make it an easy-to-use tool for many types of annotations. Even if the first prototype of CAT has been used to perform temporal and event annotation following the It-TimeML specifications, the tool is general enough to be used for annotating a broad range of linguistic and semantic phenomena. CAT is freely available for research purposes.

Keywords: annotation tool, corpus annotation, TimeML

1. Introduction

The CELCT Annotation Tool, hereafter mentioned with the acronym CAT, is a general-purpose web-based text annotation tool developed by the Center for the Evaluation of Language and Communication Technologies (CELCT). The aim of the tool is to make the annotation activity as intuitive as possible, supplying, at the same time, a rich set of features. The main strengths of CAT are flexibility, practical usability and customizability. It also supports multi-layer annotation in order to combine the annotation of several linguistic/semantic layers, it provides a standardized XML stand-off output format to help convertibility, and it contains useful features such as a facility for searching in files and one for measuring the inter-annotator agreement between two or more annotators. Moreover, CAT is freely available for research purposes.

The first prototype of CAT has been used for temporal and event annotation following the It-TimeML specifications (Caselli et al., 2011). This annotation scenario will be illustrated in Section 4 after giving an overview of similar tools for text annotation (Section 2) and after describing the main functionalities and the structure of CAT (Section 3). This first use of CAT does not mean that the tool is limited to a type of annotation of this kind. On the contrary, the tool is enough general and powerful to be used to annotate a broad range of linguistic and semantic phenomena. Finally, conclusions and future work will be presented in Section 5.

2. Related Work

The need for richly annotated corpora has been progressively increasing over the years. A corpus in a machine-readable form enriched with annotations is essential in order to automatically analyze linguistic phenomena at various levels (from morphology to syntax, from semantics to pragmatics) and can be used in many fields, such as sociology, psychology, cultural and historical studies. Moreover, annotated corpora are crucial for supervised machine learning methods in order to develop and evaluate NLP technologies.

In this context, several tools have been developed with the aim to simplify and speed up manual text annotation, which is a complex, expensive and time-consuming activity. In general, two types of tools can be distinguished: custom tools, specific for a task or a project, and general-purpose annotation tools, adaptable to different tasks. The former are not flexible: it's hard or even impossible to adapt them to annotation tasks and formats other than the ones which they were initially designed for. Examples of custom tools are: LabelMe[1] for image annotation (Russell et al., 2005), Tango[2], for temporal annotation (Verhagen et al., 2006), Jubilee and Cornerstone[3] for the annotation of Propbank (Choi et al., 2010a; Choi et al., 2010b). On the other hand, adaptability and interoperability among different annotation tasks and formats are the strong points of general-purpose annotation tools. Some examples of this kind of tools are: Callisto[4], Knowtator[5] (Ogren, 2006), MMAX2[6] (Müller and Strube, 2001) and CLaRK[7] (Simov et al., 2001).

A further distinction can be drawn between stand-alone and web-based tools. Among the tools previously mentioned, only LabelMe is web-based but, at the present time, there is a growing interest in this area: indeed, in addition to the Brandeis Annotation Tool (BAT; Verhagen, 2010), brat[8] and Slate (Kaplan et al., 2012) are two web-based tools currently under release.

Following these categorizations, CAT can be classified as a general-purpose web-based text annotation tool and a

[1] http://labelme.csail.mit.edu/
[2] http://timeml.org/site/tango/tool.html
[3] http://code.google.com/p/propbank/
[4] http://callisto.mitre.org
[5] http://knowtator.sourceforge.net/index.shtml
[6] http://mmax2.sourceforge.net/
[7] http://www.bultreebank.org/clark/
[8] http://brat.nlplab.org/

comparison can be made with the similar tools cited above. For example, with respect to Callisto, in CAT relations between discontinuous portions of text can be easily annotated. Moreover, the user can define the annotation schema through an intuitive graphical interface without having to draw on previously created ontologies as in Knowtator or on a DTD as in MMAX and Callisto. This feature makes CAT independent from other tools (Knowtator is a plug-in of Protégé (Noy, 2003)) and is useful in particular when working with a not stable format, as often happens at the beginning of a project. Finally, compared with CLaRK, CAT does not require programming skills in order to install and use it but it is ready to use.
The main features of CAT are detailed in the next section.

3. CAT Description

3.1 Features

One of the main motivations behind CAT was to develop a tool capable of supporting human annotators in order to perform linguistic and semantic text annotation within a reasonable amount of time. Indeed, manual annotation is a time consuming activity but, in practice, work is almost always subject to strict project deadlines. To overcome this difficulty, a number of features have been implemented in CAT that make it a versatile and easy-to-use tool for many types of annotations. These features are related to simplicity, customizability, multi-layer annotation, quality assurance, interoperability and convertibility (Dipper et al., 2004).
First of all, CAT is a ready-to-use tool. The installation does not require advanced computing knowledge and the pre-processing of source data is optional: users can annotate raw text files or, as an alternative, import a tokenized file.
Special effort was devoted to make CAT a simple and user-friendly tool: CAT's intuitive and easy-to-use interface allows users with little or no prior knowledge about annotation tools to perform their work. For example, highlighting features are available to visualize the annotated information and users can modify the color of annotated elements and change font size at any time. Moreover, help messages are displayed when passing with the mouse over interface elements, a search facility is integrated in the tool and for all the main actions there are keyboard shortcuts.
Customizability is essential: general-purpose annotation tools should be adaptable to new tasks and flexible to allow changes within an already defined annotation schema because, especially at the beginning of a project, tagset definitions may change quite often. For this reason, CAT has been developed to easily define and perform many types of annotations. Annotation tasks (e.g. Chunking, Named Entities, TimeML), markables, relations among markables, attributes and values for each annotation task can be defined, modified and deleted at any time to meet different annotation needs through an easy-to-use graphical user interface.
CAT enables multi-layer annotations to be performed. One or more annotation layers together form an annotation task and different annotation tasks can be executed on the same file. Each layer is identified by the type of annotated markable. Therefore it is possible to annotate overlapped portions of text and to dynamically show or hide an entire layer annotated with a specific type of markable. A layer can also contain *empty tags*, that is tags with no textual content, useful in many annotation tasks (e.g. annotation of anaphora, ellipsis, TimeML).
As regards quality assurance, in order to achieve a consistent annotation, a DTD is automatically generated on the basis of markable and relation definitions made by the users within the tool: on the basis of this DTD, the syntax is checked and errors are notified when exporting the corpus. In addition, an inter-annotator agreement statistics system is integrated in CAT which helps to have an high-quality annotation: in particular, this built-in facility allows to measure the Dice's coefficient for markable extents and relation detection and Cohen's or Fleiss' Kappa for attributes values of markables and relations (Artstein and Poesio, 2008). Another quality control method is indirectly given by the fact that a group of annotators working on the same project can share the account allowing to monitor the work and perform collaborative annotation of the same files.
The tool supports UTF-8 encoding and provides an XML-based stand-off format as output. In the CAT stand-off format different annotation layers are contained in separate document sections and are related to each other and to source data through pointers. This type of standardized output facilitates data exchange and allows users to convert annotated texts into other formats. Furthermore, given that UTF-8 can encode any Unicode character, texts in different languages and with special characters can be easily displayed.

3.2 Technical Details

From the technical point of view, the core idea on which CAT is based is to reproduce the basic concepts of object oriented programming. Each element of the annotation (e.g. markables, attributes, relations) is a particular instance of an object category. Each element has a list of properties and a list of *methods* usable by other objects in order to build a sort of "net" that represents a possible annotation schema. Thanks to this approach it is possible to define a large number of markables and relations because each of these two elements is a combination of other instances defined by the user.
CAT's object oriented approach is strictly based on the macro-categorization of the general concept of textual annotation interpreted as a chain of simple operations (e.g. text delimitation, property attributions, etc.). On the basis of these operations there is the identification of the unit of text to be used as the Minimum Markable Unit (MMU). A MMU can be, for example, a single character, a word or a longer portion of text. Each MMU can be associated with an annotation label taken from a pre-defined annotation schema. Different annotation schema may adopt different

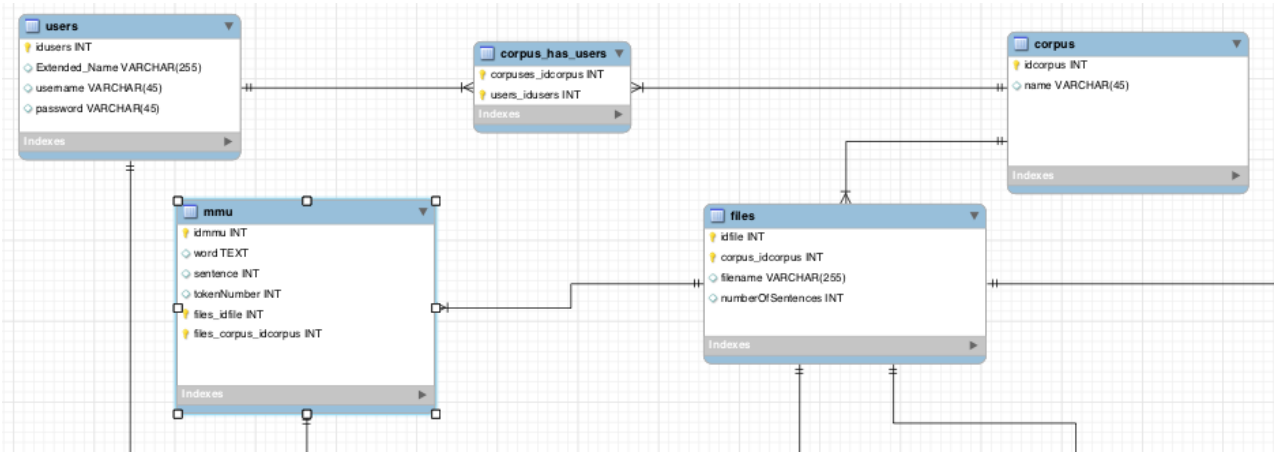

Figure 1: Subsection of the relation schema

MMUs. For instance, the annotation of temporal expressions can be performed on a single-word basis, e.g. "<Yesterday>, I went to the restaurant with my two English teachers". Conversely, the annotation of a poem can assume a complete line as the MMU, e.g. "<From off a hill whose concave womb reworded>". Most of the currently available annotation tools adopt by default a single character as MMU. CAT allows users to specify the extent of the MMU for each annotation project. When working on the text, users can, by a single click, recall a MMU to be annotated, instead of manually highlighting a selected portion of text for coding. This can substantially fasten the annotation process, especially in cases where MMUs are particularly long (e.g. poem). Further, annotation is more precise, as the risk of selecting incomplete units is reduced to a minimum.

While each project assumes the pre-defined MMU as the basis for the annotation, nested annotations are also allowed. Portions of each MMU can be selected and annotated independently. The original MMU will not be split, but additional annotations will be mapped onto the original tagged MMU. For example, if in an annotation task the predefined MMU is the token, the compound word "lifetime" can be annotated as a single unit but also by identifying two sub-sections within it, i.e. "life" and "time", mapped onto the original MMU.

In the initial stages of the development of the tool, it was necessary to build an efficient relation schema that could suit most application scenarios. Figure 1 shows a simplified representation of the portion of the relation schema describing the relationship among users, corpus, files and MMUs.

The main elements involved in the annotation process and defined in the relation schema are markables, relations and attributes. A markable is always identified by a label, can be anchored to the text or not (as in case of empty tags) and can have one or more attributes. An attribute is a simple key-value pair and can be associated to a markable or a relation. A relation is an element that bridges two or more markables. A relation has at least two properties: directionality (from source to target, from target to source, bidirectional) and cardinality (one-to-one, many-to-one, one-to-many, many-to-many).

The next step in the realization of the tool, was the creation an interface that allows the user to interact with all the various elements of the relation schema mentioned above. To manage the graphical interface through the web browser, different techniques have been used. In particular, interface events, such as mouse click, drag 'n' drop, hot key detection, have been realized using the JQuery JavaScript library. This set of instructions provides a sort of "asynchronous" bridge between the Java backend (servlet) and the graphical frontend. JQuery library offers a very easy to use collection of methods and functions for DOM manipulation and data structure analysis.

As mentioned in Section 3.1, CAT offers the possibility of multi-layer annotation. From the graphical point of view the layer mechanism has been implemented using a sort of "Chinese-box" encapsulation: in the web interface, a MMU is an aggregation of html tags that identifies the type of annotation and the graphical layering in the page rendering.

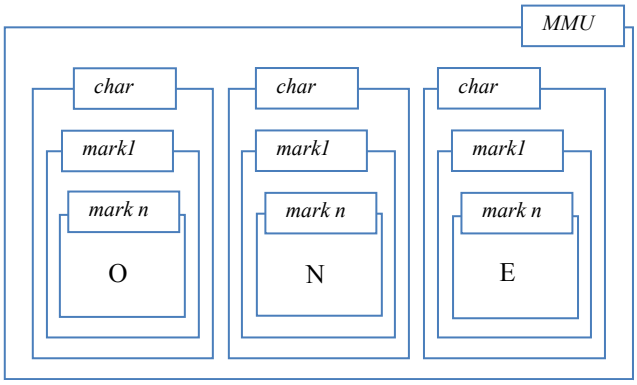

Figure 2: "Chinese-box" encapsulation for multi-layer annotation

Figure 2 shows the representation of the MMU "one". Each character is identified by Figure 2 shows the representation of the MMU "one" from the graphical point of view. Each character is identified by a macro-CSS-class "char" and each class "char" contains self-encapsulated micro-CSS-classes that represent the annotation of the markables.

The graphical interface of CAT is interconnected with a Tomcat application server. The data exchange between the GUI and the application server has been realized with the *JavaScript Object Notation* (JSON). This solution, very efficient and easy to use, reflects the concept of "entities / objects" trading treated in our approach. Data are stored in a MySQL DataBase. The database access is delegated to

the servlet engine of the application server; this solution grants a robust and multi-thread processing of requests and a greater security during data transaction.
In Figure 3, the three macro-levels of software stratification are represented.

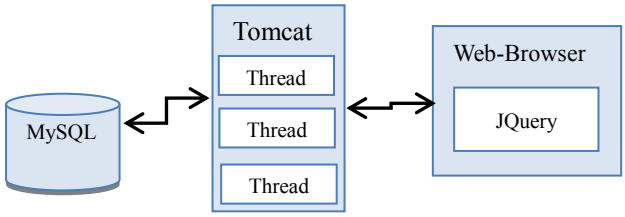

Figure 3: Software stratification from data storage to graphical user interface

4. An Example of CAT in Use: the IT-TimeML Annotation Experience

TimeML (Pustejovsky et al., 2005) is a formalism which is focused on *Events* (i.e. actions, states, and processes - <EVENT> tag), *Temporal Expressions* (i.e. durations, calendar dates, times of day and sets of time - <TIMEX3> tag), *Signals* (e.g. temporal prepositions and subordinators - <SIGNAL> tag) and various kind of *dependencies between Events and/or Temporal Expressions* (i.e. temporal, aspectual and subordination relations - <TLINK>, <ALINK> and <SLINK> tags respectively).

The success of this annotation schema promoted the growth of the interest in temporal processing in the NLP community and, over the years, several annotation tools (both automatic and manual) and annotated corpora have been developed for this specific task. In particular, the ISO TC 37 / SC 4 initiative ("Terminology and other language and content resources") and the TempEval contests in 2007 and 2010 (Verhagen, 2007; Pustejovsky, 2010) have contributed to the development of TimeML compliant annotation schemata and corpora in different languages, such as Spanish, Korean, Chinese and French.
In this context, CELCT worked within the LiveMemories[9] project on the creation of a new semantic resource for Italian called *CELCT Corpus* which is part of the Ita-TimeBank, the largest Italian corpus annotated with information for temporal processing following the TimeML guidelines for Italian (It-TimeML; Caselli, 2010). In order to perform the TimeML annotation, two possibilities were available: the combination of Callisto and Tango, or the use of the Brandeis Annotation Tool. Annotating with two different tools, as in the former possibility, is very time consuming and the visualization and manipulation of links in Tango can become unclear when a large number of relations has been annotated on the same text. On the other hand, BAT is an intuitive web tool that provides many features; in particular, it allows to control the parallel annotation of the same texts when many annotators are available. However, when our project started, BAT was still in progress and not all TimeML annotation

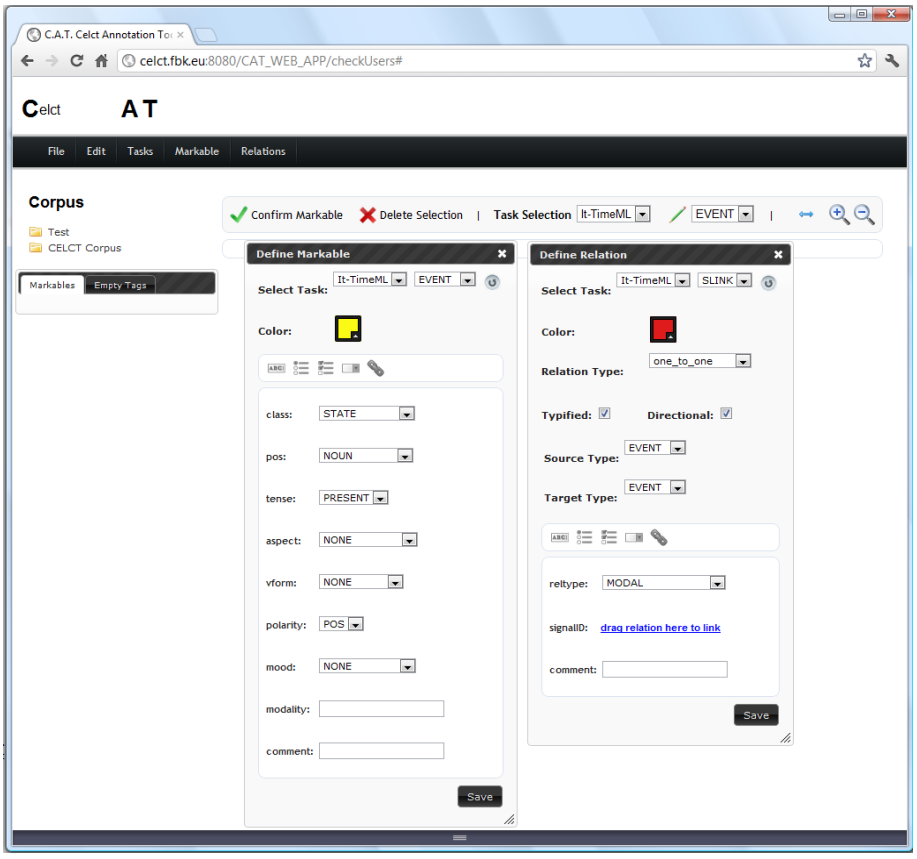

Figure 4: Definition of the <EVENT> markable and of the <SLINK> relation

[9] http://www.livememories.org/

layers were available (e.g. SLINK and ALINK). Moreover, the tool did not allow to have a unitary view of all markables and to easily modify the annotations made in an annotation layer. Considering all these motivations, we decided to test the first prototype of CAT on the It-TimeML annotation.

The first step was the definition of It-TimeML markables and relations. CAT allows to choose the name of each tag and relation, the annotation highlight color and all the attributes using five types of controls, namely i) Text Box Element; ii) Radio Button Element; iii) Check Box Element; iv) Dropdown Menu Element; v) Annotation Reference Element. Relations have also other attributes to be defined, such as the cardinality and the directionality.

Figure 4 shows the definition of the tag for Events and of the subordination relation within the tool using intuitive graphical interfaces. The name of the tag is "EVENT", the annotation highlight color is yellow and it has nine attributes: seven are defined as dropdown menu elements and two as text box elements. As regards the subordination relation, the name of the link is "SLINK", the annotation highlight color is red, it is a one-to-one relation, it is typified (both arguments are Events) and directional. Finally, three attributes are defined: one as dropdown menu element, one as annotation reference element, and one as text box element.

The second step was the actual annotation of the corpus. Portions of text to be annotated were selected and related to each markable type. Empty tags were added when necessary. Attribute values were assigned by filling in a form that pops up after clicking on the tag extent. Annotated elements were then selected as relation arguments and a specific form was used to assign attribute values to relations.

The final outcome is presented in Figure 5: Events, Temporal Expressions and Signals are highlighted with different colors, the list of relations, organized by type, is displayed under the text and a popup window shows the attributes of an aspectual link. Moreover, an empty tag anchored to a temporal expression is shown in a dedicated panel on the left.

Finally, after completing the annotation process, annotated files were exported in a It-TimeML compliant XML format.

In conclusion, CAT has been used to annotate the *CELCT Corpus* which consists of more than 180,000 tokens annotated with Temporal Expressions and more than 90,000 tokens annotated also with Events, Signals and Links (see Table 1 for details).

Markables & Relations	#
TIMEX3	4,852
EVENT	17,554
SIGNAL	2,045
TLINK	3,373
SLINK	3,985
ALINK	238

Table 1: Number of annotated markables and relations in the *CELCT Corpus*

For what concern the effort, the annotation of the *CELCT Corpus* required 1.3 person/years and it was performed achieving inter-coder agreement scores comparable or higher than the ones obtained in the annotation of the English TimeBank 1.2 (Pustejovsky et al., 2006).

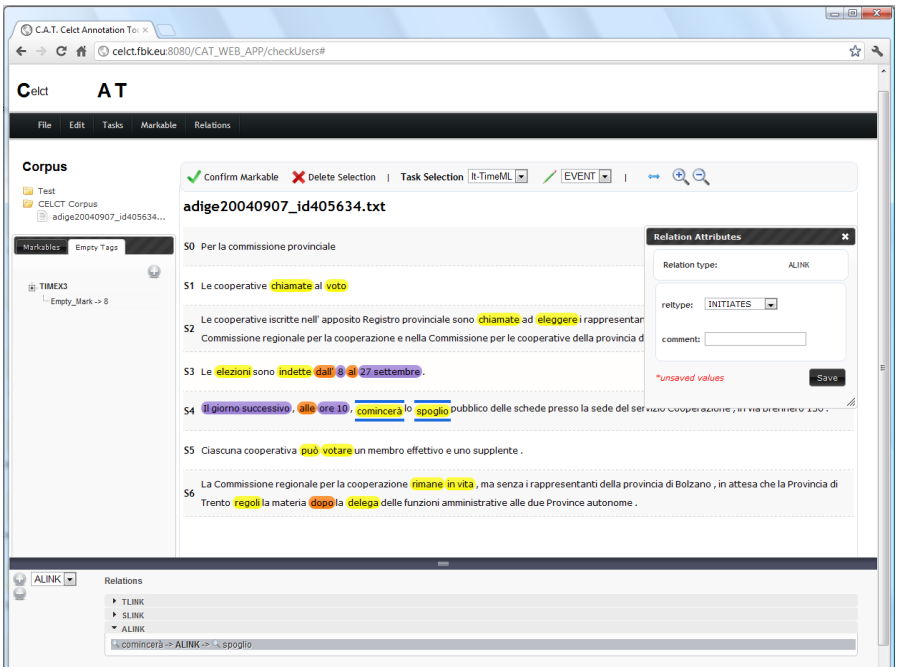

Figure 5: An annotated file in CAT

5. Conclusions and Future Work

In this paper a new general-purpose web-based textual annotation tool called CAT has been presented and its main features have been described. Since now the tool has been used in the LiveMemories project for the annotation of temporal and event information in Italian texts following the It-TimeML specifications. This real annotation scenario has been described also providing figures to visually explain the practical use of the tool.

Recently, CAT has been chosen to perform semantic annotation of stories for children within the TERENCE European project[10]. In addition, it will be used for the annotation of modalised and negated events within the "Processing Modality and Negation" pilot task in the QA4MRE exercise of the CLEF 2012 evaluation campaign[11].

For what concerns future work, we are going to release a local CAT application creating a simple installation package. The package will provide a custom installation of Tomcat running on a non-standard port and, for data storage, this local application of CAT will use SQLite DB engine.

In addition, some features and extensions can be designed and implemented. For example, we plan to give the possibility to add media types (e.g. image, video, audio, etc.) as attributes and to incorporate mechanisms for the on-the-fly checking of annotation well-formedness (e.g. verifying the syntax of an attribute value through patterns).

6. References

Artstein, R., Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics,* 34(4), pp. 555--596.

Caselli, T., Bartalesi Lenzi, V., Sprugnoli, R., Pianta, E., Prodanof, I. (2011). Annotating Events, Temporal Expressions and Relations in Italian: the It-TimeML Experience for the Ita-TimeBank. In *Proceedings of LAW V*. Portland, Oregon.

Caselli, T. (2010). It-TimeML: TimeML Annotation Scheme for Italian, Version 1.3.1. *Technical Report*.

Choi, J. D., Bonial, C., Palmer, M. (2010a). PropBank instance annotation guidelines using a dedicated editor, Jubilee. In *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC 2010)*, pp. 1871--1875.

Choi, J. D., Bonial, C., Palmer, M. (2010b). PropBank FrameSet annotation guidelines using a dedicated editor, Cornerstone. In *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC 2010)*, pp. 3650--3653.

Dipper, S., Götze, M., Stede, M. (2004). Simple annotation tools for complex annotation tasks: an evaluation. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, pp. 54--62. Lisbon, Portugal.

Kaplan, D., Iida, R., Nishina, K., Tokunaga, T. (2012). Slate – A Tool for Creating and Maintaining Annotated Corpora. *Journal for Language Technology and Computational Linguistics*, 26(2), pp. 89--101.

Müller, C., Strube, M. (2001). MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Seattle, WA.

Noy, N.F., Crubezy, M., Fergerson, R.W., Knublauch, H., Tu, S.W., Vendetti, J., Musen, M.A. (2003). Protégé-2000: An open-source Ontology-development and Knowledge-Acquisition Environment. In *Proceedings of AMIA Annual Symposium*.

Ogren, P., (2006). Knowtator: A Protégé plug-in for annotated corpus construction. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. ACM Press.

Pustejovsky, J., Knippen, R., Littman, J., Saurí, R. (2005). Temporal and Event Information in Natural language Text. *Language Resources and Evaluation*, 39(2-3), pp. 123--164.

Pustejovsky, J., Littman, J., Saurí, R., Verhagen, M. (2006). TimeBank 1.2 Documentation. http://timeml.org/site/timebank/documentation-1.2.html

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2005). Labelme: A database and web-based tool for image annotation. *Technical report*, Massachusetts Institute of Technology.

Simov, K., Peev, Z., Kouylekov, M., Simov, A., Dimitrov, M., Kiryakov, A. (2001). CLaRK - an XML-based System for Corpora Development. In *Proceedings of the Corpus Linguistics 2001 Conference*. Lancaster, UK.

Verhagen, M., Knippen, R., Mani, I., and Pustejovsky, J. (2006). Annotation of Temporal Relations with Tango. In *Proceedings of Language Resources and Evaluation Conference (LREC 2006)*. Genoa, Italy.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, J. (2007). SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*. Prague, Czech Republic.

Pustejovsky J., Verhagen, M. (2010). SemEval-2010 Task 13: Evaluating Events, Time Expressions, and Temporal Relations (TempEval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Boulder, CO.

Verhagen, M. (2010) The Brandeis Annotation Tool. In *Proceedings of Language Resources and Evaluation Conference (LREC 2010)*. Valletta, Malta.

[10] http://terenceproject.eu/

[11] http://clef2012.org/