

BRAT: a Web-based Tool for NLP-Assisted Text Annotation

Pontus Stenetorp^{1*} Sampo Pyysalo^{2,3*} Goran Topić¹
Tomoko Ohta^{1,2,3} Sophia Ananiadou^{2,3} and Jun'ichi Tsujii⁴

¹Department of Computer Science, The University of Tokyo, Tokyo, Japan

²School of Computer Science, University of Manchester, Manchester, UK

³National Centre for Text Mining, University of Manchester, Manchester, UK

⁴Microsoft Research Asia, Beijing, People's Republic of China

{pontus, smp, goran, okap}@is.s.u-tokyo.ac.jp

sophia.ananiadou@manchester.ac.uk

jtsujii@microsoft.com

Abstract

We introduce the brat rapid annotation tool (BRAT), an intuitive web-based tool for text annotation supported by Natural Language Processing (NLP) technology. BRAT has been developed for rich structured annotation for a variety of NLP tasks and aims to support manual curation efforts and increase annotator productivity using NLP techniques. We discuss several case studies of real-world annotation projects using pre-release versions of BRAT and present an evaluation of annotation assisted by semantic class disambiguation on a multi-category entity mention annotation task, showing a 15% decrease in total annotation time. BRAT is available under an open-source license from: <http://brat.nlplab.org>

1 Introduction

Manually-curated gold standard annotations are a prerequisite for the evaluation and training of state-of-the-art tools for most Natural Language Processing (NLP) tasks. However, annotation is also one of the most time-consuming and financially costly components of many NLP research efforts, and can place heavy demands on human annotators for maintaining annotation quality and consistency. Yet, modern annotation tools are generally technically oriented and many offer little support to users beyond the minimum required functionality. We believe that intuitive and user-friendly interfaces as well as the judicious application of NLP technology to *support*, not *supplant*, human judgements can help maintain the quality of annotations, make annotation more accessible to non-technical users such as subject

*These authors contributed equally to this work

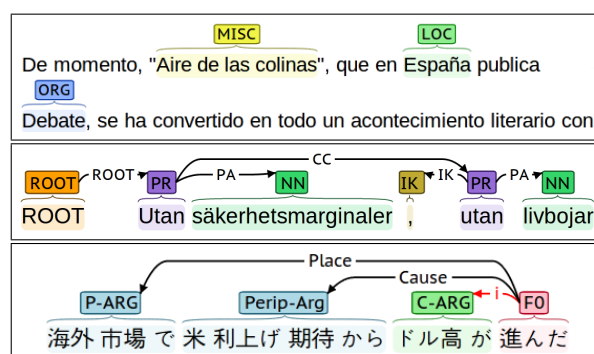


Figure 1: Visualisation examples. Top: named entity recognition, middle: dependency syntax, bottom: verb frames.

domain experts, and improve annotation productivity, thus reducing both the human and financial cost of annotation. The tool presented in this work, BRAT, represents our attempt to realise these possibilities.

2 Features

2.1 High-quality Annotation Visualisation

BRAT is based on our previously released open-source STAV text annotation visualiser (Stenetorp et al., 2011b), which was designed to help users gain an understanding of complex annotations involving a large number of different semantic types, dense, partially overlapping text annotations, and non-projective sets of connections between annotations. Both tools share a vector graphics-based visualisation component, which provide scalable detail and rendering. BRAT integrates PDF and EPS image format export functionality to support use in e.g. figures in publications (Figure 1).

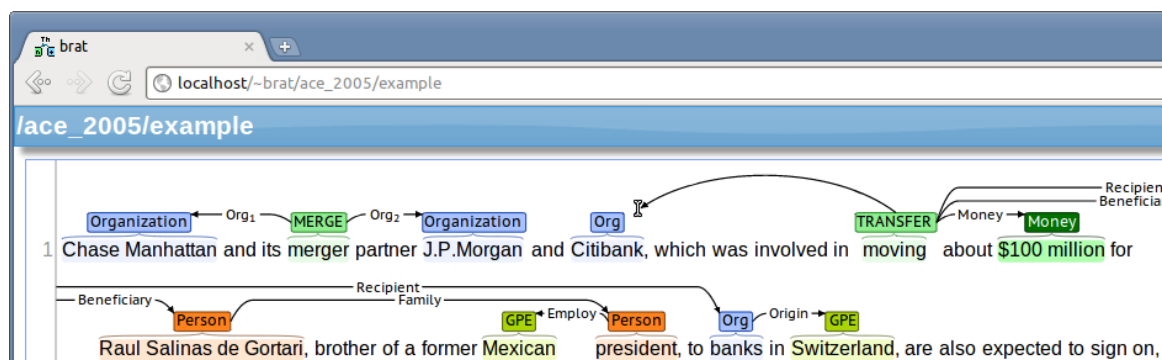


Figure 2: Screenshot of the main BRAT user-interface, showing a connection being made between the annotations for “moving” and “Citibank”.

2.2 Intuitive Annotation Interface

We extended the capabilities of STAV by implementing support for annotation editing. This was done by adding functionality for recognising standard user interface gestures familiar from text editors, presentation software, and many other tools.

In BRAT, a span of text is marked for annotation simply by selecting it with the mouse by “dragging” or by double-clicking on a word. Similarly, annotations are linked by clicking with the mouse on one annotation and dragging a connection to the other (Figure 2).

BRAT is browser-based and built entirely using standard web technologies. It thus offers a familiar environment to annotators, and it is possible to start using BRAT simply by pointing a standards-compliant modern browser to an installation. There is thus no need to install or distribute any additional annotation software or to use browser plug-ins. The use of web standards also makes it possible for BRAT to uniquely identify any annotation using Uniform Resource Identifiers (URIs), which enables linking to individual annotations for discussions in e-mail, documents and on web pages, facilitating easy communication regarding annotations.

2.3 Versatile Annotation Support

BRAT is fully configurable and can be set up to support most text annotation tasks. The most basic annotation primitive identifies a text span and assigns it a type (or tag or label), marking for e.g. POS-tagged tokens, chunks or entity mentions (Figure 1 top). These base annotations can be connected by binary relations – either directed or undirected – which can be configured for e.g. simple relation extraction, or verb frame annotation

(Figure 1 middle and bottom). n -ary associations of annotations are also supported, allowing the annotation of event structures such as those targeted in the MUC (Sundheim, 1996), ACE (Doddington et al., 2004), and BioNLP (Kim et al., 2011) Information Extraction (IE) tasks (Figure 2). Additional aspects of annotations can be marked using *attributes*, binary or multi-valued flags that can be added to other annotations. Finally, annotators can attach free-form text notes to any annotation.

In addition to information extraction tasks, these annotation primitives allow BRAT to be configured for use in various other tasks, such as chunking (Abney, 1991), Semantic Role Labeling (Gildea and Jurafsky, 2002; Carreras and Màrquez, 2005), and dependency annotation (Nivre, 2003) (See Figure 1 for examples). Further, both the BRAT client and server implement full support for the Unicode standard, which allow the tool to support the annotation of text using e.g. Chinese or Devanāgarī characters. BRAT is distributed with examples from over 20 corpora for a variety of tasks, involving texts in seven different languages and including examples from corpora such as those introduced for the CoNLL shared tasks on language-independent named entity recognition (Tjong Kim Sang and De Meulder, 2003) and multilingual dependency parsing (Buchholz and Marsi, 2006).

BRAT also implements a fully configurable system for checking detailed constraints on annotation semantics, for example specifying that a TRANSFER event must take exactly one of each of GIVER, RECIPIENT and BENEFICIARY arguments, each of which must have one of the types PERSON, ORGANIZATION or GEO-POLITICAL ENTITY, as well as a MONEY argument of type

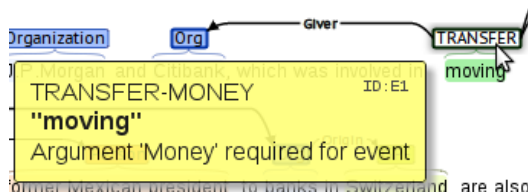


Figure 3: Incomplete TRANSFER event indicated to the annotator

MONEY, and may optionally take a PLACE argument of type LOCATION (LDC, 2005). Constraint checking is fully integrated into the annotation interface and feedback is immediate, with clear visual effects marking incomplete or erroneous annotations (Figure 3).

2.4 NLP Technology Integration

BRAT supports two standard approaches for integrating the results of fully automatic annotation tools into an annotation workflow: bulk annotation imports can be performed by format conversion tools distributed with BRAT for many standard formats (such as in-line and column-formatted BIO), and tools that provide standard web service interfaces can be configured to be invoked from the user interface.

However, human judgements cannot be replaced or based on a completely automatic analysis without some risk of introducing bias and reducing annotation quality. To address this issue, we have been studying ways to augment the annotation process with input from statistical and machine learning methods to support the annotation process while still involving human annotator judgement for each annotation.

As a specific realisation based on this approach, we have integrated a recently introduced machine learning-based semantic class disambiguation system capable of offering multiple outputs with probability estimates that was shown to be able to reduce ambiguity on average by over 75% while retaining the correct class in on average 99% of cases over six corpora (Stenetorp et al., 2011a). Section 4 presents an evaluation of the contribution of this component to annotator productivity.

2.5 Corpus Search Functionality

BRAT implements a comprehensive set of search functions, allowing users to search document col-

Figure 4: The BRAT search dialog

lections for text span annotations, relations, event structures, or simply text, with a rich set of search options definable using a simple point-and-click interface (Figure 4). Additionally, search results can optionally be displayed using keyword-in-context concordancing and sorted for browsing using any aspect of the matched annotation (e.g. type, text, or context).

3 Implementation

BRAT is implemented using a client-server architecture with communication over HTTP using JavaScript Object Notation (JSON). The server is a RESTful web service (Fielding, 2000) and the tool can easily be extended or adapted to switch out the server or client. The client user interface is implemented using XHTML and Scalable Vector Graphics (SVG), with interactivity implemented using JavaScript with the jQuery library. The client communicates with the server using Asynchronous JavaScript and XML (AJAX), which permits asynchronous messaging.

BRAT uses a stateless server back-end implemented in Python and supports both the Common Gateway Interface (CGI) and FastCGI protocols, the latter allowing response times far below the 100 ms boundary for a “smooth” user experience without noticeable delay (Card et al., 1983). For server side annotation storage BRAT uses an easy-to-process file-based stand-off format that can be converted from or into other formats; there is no need to perform database import or export to interface with the data storage. The BRAT server in-

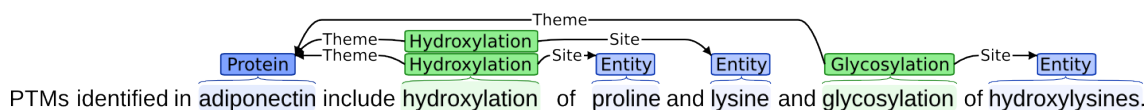


Figure 5: Example annotation from the BioNLP Shared Task 2011 Epigenetics and Post-translational Modifications event extraction task.

stallation requires only a CGI-capable web server and the set-up supports any number of annotators who access the server using their browsers, on any operating system, without separate installation.

Client-server communication is managed so that all user edit operations are immediately sent to the server, which consolidates them with the stored data. There is no separate “save” operation and thus a minimal risk of data loss, and as the authoritative version of all annotations is always maintained by the server, there is no chance of conflicting annotations being made which would need to be merged to produce an authoritative version. The BRAT client-server architecture also makes real-time collaboration possible: multiple annotators can work on a single document simultaneously, seeing each others edits as they appear in a document.

4 Case Studies

4.1 Annotation Projects

BRAT has been used throughout its development during 2011 in the annotation of six different corpora by four research groups in efforts that have in total involved the creation of well-over 50,000 annotations in thousands of documents comprising hundreds of thousands of words.

These projects include structured event annotation for the domain of cancer biology, Japanese verb frame annotation, and gene-mutation-phenotype relation annotation. One prominent effort making use of BRAT is the BioNLP Shared Task 2011,¹ in which the tool was used in the annotation of the EPI and ID main task corpora (Pyysalo et al., 2012). These two information extraction tasks involved the annotation of entities, relations and events in the epigenetics and infectious diseases subdomains of biology. Figure 5 shows an illustration of shared task annotations.

Many other annotation efforts using BRAT are still ongoing. We refer the reader to the BRAT

¹<http://2011.bionlp-st.org>

Mode	Total	Type Selection
Normal	45:28	13:49
Rapid	39:24 (-6:04)	09:35 (-4:14)

Table 1: Total annotation time, portion spent selecting annotation type, and absolute improvement for rapid mode.

website² for further details on current and past annotation projects using BRAT.

4.2 Automatic Annotation Support

To estimate the contribution of the semantic class disambiguation component to annotation productivity, we performed a small-scale experiment involving an entity and process mention tagging task. The annotation targets were of 54 distinct mention types (19 physical entity and 35 event/process types) marked using the simple typed-span representation. To reduce confounding effects from annotator productivity differences and learning during the task, annotation was performed by a single experienced annotator with a Ph.D. in biology in a closely related area who was previously familiar with the annotation task.

The experiment was performed on publication abstracts from the biomolecular science subdomain of glucose metabolism in cancer. The texts were drawn from a pool of 1,750 initial candidates using stratified sampling to select pairs of 10-document sets with similar overall statistical properties.³ Four pairs of 10 documents (80 in total) were annotated in the experiment, with 10 in each pair annotated with automatic support and 10 without, in alternating sequence to prevent learning effects from favouring either approach.

The results of this experiment are summarized in Table 1 and Figure 6. In total 1,546 annotations were created in normal mode and 1,541 annota-

²<http://brat.nlplab.org>

³Document word count and expected annotation count, were estimated from the output of NERSuite, a freely available CRF-based NER tagger: <http://nersuite.nlplab.org>

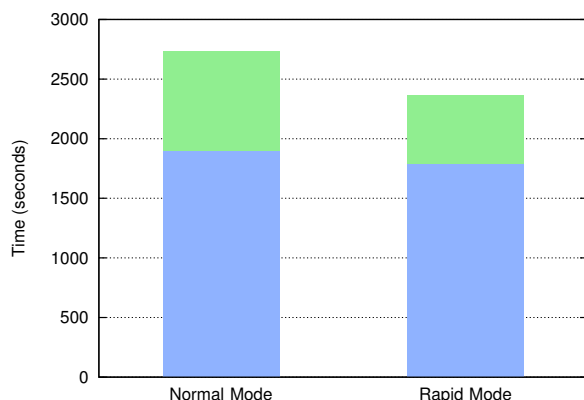


Figure 6: Allocation of annotation time. GREEN signifies time spent on selecting annotation type and BLUE the remaining annotation time.

tions in rapid mode; the sets are thus highly comparable. We observe a 15.4% reduction in total annotation time, and, as expected, this is almost exclusively due to a reduction in the time the annotator spent selecting the type to assign to each span, which is reduced by 30.7%; annotation time is otherwise stable across the annotation modes (Figure 6). The reduction in the time spent in selecting the span is explained by the limiting of the number of candidate types exposed to the annotator, which were decreased from the original 54 to an average of 2.88 by the semantic class disambiguation component (Stenetorp et al., 2011a).

Although further research is needed to establish the benefits of this approach in various annotation tasks, we view the results of this initial experiment as promising regarding the potential of our approach to using machine learning to support annotation efforts.

5 Related Work and Conclusions

We have introduced BRAT, an intuitive and user-friendly web-based annotation tool that aims to enhance annotator productivity by closely integrating NLP technology into the annotation process. BRAT has been and is being used for several ongoing annotation efforts at a number of academic institutions and has so far been used for the creation of well-over 50,000 annotations. We presented an experiment demonstrating that integrated machine learning technology can reduce the time for type selection by over 30% and overall annotation time by 15% for a multi-type entity mention annotation task.

The design and implementation of BRAT was

informed by experience from several annotation tasks and research efforts spanning more than a decade. A variety of previously introduced annotation tools and approaches also served to guide our design decisions, including the fast annotation mode of Knowtator (Ogren, 2006), the search capabilities of the XConc tool (Kim et al., 2008), and the design of web-based systems such as MyMiner (Salgado et al., 2010), and GATE Teamware (Cunningham et al., 2011). Using machine learning to accelerate annotation by supporting human judgements is well documented in the literature for tasks such as entity annotation (Tsuruoka et al., 2008) and translation (Martínez-Gómez et al., 2011), efforts which served as inspiration for our own approach.

BRAT, along with conversion tools and extensive documentation, is freely available under the open-source MIT license from its homepage at <http://brat.nlplab.org>

Acknowledgements

The authors would like to thank early adopters of BRAT who have provided us with extensive feedback and feature suggestions. This work was supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan), the UK Biotechnology and Biological Sciences Research Council (BB-SRC) under project Automated Biological Event Extraction from the Literature for Drug Discovery (reference number: BB/G013160/1), and the Royal Swedish Academy of Sciences.

References

- Steven Abney. 1991. Parsing by chunks. *Principle-based parsing*, 44:257–278.
- Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164.
- Stuart K. Card, Thomas P. Moran, and Allen Newell. 1983. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic Role Labeling. In *Proceedings of the 9th Conference on Natural Language Learning*, pages 152–164. Association for Computational Linguistics.
- Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) program: Tasks, data, and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 837–840.
- Roy Fielding. 2000. REpresentational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine, page 120.
- Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.
- Jin-Dong Kim, Tomoko Ohta, and Jun’ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10.
- Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun’ichi Tsujii. 2011. Overview of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, June. Association for Computational Linguistics.
- LDC. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events. Technical report, Linguistic Data Consortium.
- Pascual Martínez-Gómez, Germán Sanchis-Trilles, and Francisco Casacuberta. 2011. Online learning via dynamic reranking for computer assisted translation. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 6609 of *Lecture Notes in Computer Science*, pages 93–105. Springer Berlin / Heidelberg.
- Joakim Nivre. 2003. An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 149–160.
- Philip V. Ogren. 2006. Knowtator: A protégé plug-in for annotated corpus construction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Companion Volume: Demonstrations*, pages 273–275, New York City, USA, June. Association for Computational Linguistics.
- Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Junichi Tsujii, and Sophia Ananiadou. 2012. Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011. *BMC Bioinformatics*, 13(suppl. 8):S2.
- David Salgado, Martin Krallinger, Marc Depaule, Elodie Drula, and Ashish V Tendulkar. 2010. Myminer system description. In *Proceedings of the Third BioCreative Challenge Evaluation Workshop 2010*, pages 157–158.
- Pontus Stenetorp, Sampo Pyysalo, Sophia Ananiadou, and Jun’ichi Tsujii. 2011a. Almost total recall: Semantic category disambiguation using large lexical resources and approximate string matching. In *Proceedings of the Fourth International Symposium on Languages in Biology and Medicine*.
- Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun’ichi Tsujii. 2011b. BioNLP Shared Task 2011: Supporting Resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Beth M. Sundheim. 1996. Overview of results of the MUC-6 evaluation. In *Proceedings of the Sixth Message Understanding Conference*, pages 423–442. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Yoshimasa Tsuruoka, Jun’ichi Tsujii, and Sophia Ananiadou. 2008. Accelerating the annotation of sparse named entities by dynamic sentence selection. *BMC Bioinformatics*, 9(Suppl 11):S8.