stav: text annotation visualiser

Pontus Stenetorp[*] **Goran Topić**[*] **Sampo Pyysalo**[†‡]
Sophia Ananiadou[†‡] **and Jun'ichi Tsujii**[†‡$]
[*]Department of Computer Science, University of Tokyo, Tokyo, Japan
[†]School of Computer Science, University of Manchester, Manchester, UK
[‡]National Centre for Text Mining, University of Manchester, Manchester, UK
[$]Microsoft Research Asia, Beijing, People's Republic of China
{pontus,goran,smp}@is.s.u-tokyo.ac.jp
sophia.ananiadou@manchester.ac.uk
jtsujii@microsoft.com

1 Introduction

When creating Natural Language Processing (NLP) systems and creating manually curated resources it is important for both researchers and developers to gain insight into the structure of the annotations they are working with. Intuitive visualizations of annotations are one effective way of understanding their structure. However, as the structure of annotations grows more complex, from part-of-speech tagging to chunking [1], pairwise relations [3] and n-ary associations such as event structures [4], it becomes more difficult to present annotation structure in a comprehensive and intuitive way. Yet, as the complexity of annotations grows, so does the importance of tools for making them understandable to be able to convey their nature to a wider audience.

As a part of the BioNLP 2011 Shared Task on Event Extraction (EE) [6] we created an open-source web-based visualisation tool, the STAV text annotation visualiser [8], for visualising complex n-ary associations between text-bound annotations (i.e. annotations associated with specific expressions in text). The tool was well received by the shared task participants; in particular, it proved valuable in allowing participants new to the task to better understand the structured annotations and served to make error analysis easier by enabling the visualisation of the system output, while relieving the burden on individual participants to construct visualization and analysis tools for their own systems. However, while valuable for the EE task it was originally created for, the original tool was in many aspects restricted to the specific event annotation scheme of that shared task and lacked capacity to visualize other forms of annotation.

In this work, we describe a new version of STAV that generalizes the features of the original to support a rich variety of different tasks. We also discuss use cases of visualising textual annotations as a part of an annotation effort and when performing error analysis.

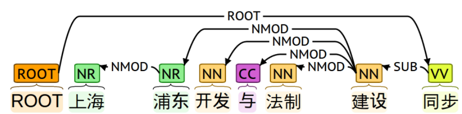

Figure 1: Dependency annotations

2 Annotation Primitives

The new generalized implementation of STAV now supports the following categories of annotation primitives:[1]

1. Typed spans, as applied e.g. in Named Entity Recognition (NER)
2. Binary relations between other annotations, as applied in relation extraction tasks
3. N-ary associations of annotations, such as event structures
4. Binary or multi-valued attributes on any annotation, e.g. NEGATION or CONFIDENCE
5. Free-form text "notes" on any annotation

As an important distinction to the previous version of the tool, all of these annotation primitives are now fully configurable in their types, values and scopes using simple text-based configuration files that allow the specification of a wide variety of different annotation tasks, including e.g. relation extraction and dependency-based syntactic annotation (as seen in Figure 1) for all of which configuration files are included with the STAV software.

[1]The original implementation of STAV lacked support for relation annotations, and attributes were restricted to binary flags on events.

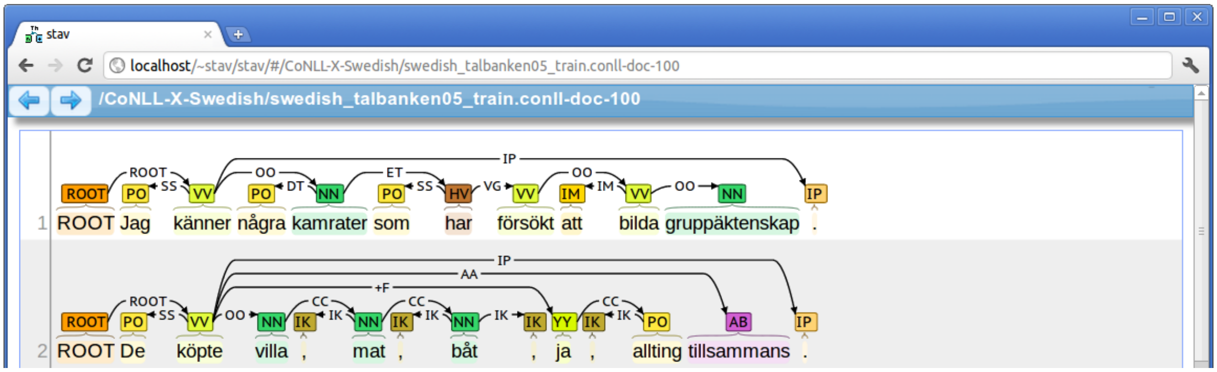

Figure 2: Screenshot of the main STAV user-interface, showing a dependency annotations from the CoNLL-X corpus [2] in the Chrome web-browser

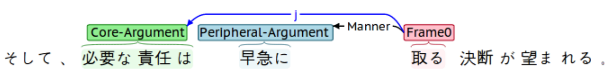

Figure 3: Verb frame annotations in Japanese

3 Features

This section introduces and discusses several key features of the STAV visualisation tool.

3.1 Extensive Language Support

While a majority of NLP research focuses on the English language, there is an increasing amount of annotated resources becoming available for other languages such as Japanese, Indian and Chinese. Since STAV is a web-based tool, incorporating support for non-ASCII characters is fairly straight-forward and a prerequisite for many non-English languages. This addition now allows STAV to handle full Unicode which greatly improves the range of languages supported by the tool.

In order to create a screen layout, the initial STAV implementation relied on spaces to separate tokens in the text. However, this assumption does not hold true for languages without explicit token markers. Although such markers can be explicitly introduced in corpora even for languages where they are not originally present [11], their availability cannot be assumed for visualisation. We therefore incorporated support for segmentation systems such as MeCab[2] and thus are now able to visualise textual annotations without requiring the annotations to contain explicit token markers (Figure 3).

3.2 Search Functionality

Search functionality serves many purposes in corpus development, analysis and error analysis. To assist with these tasks we implemented text search as well

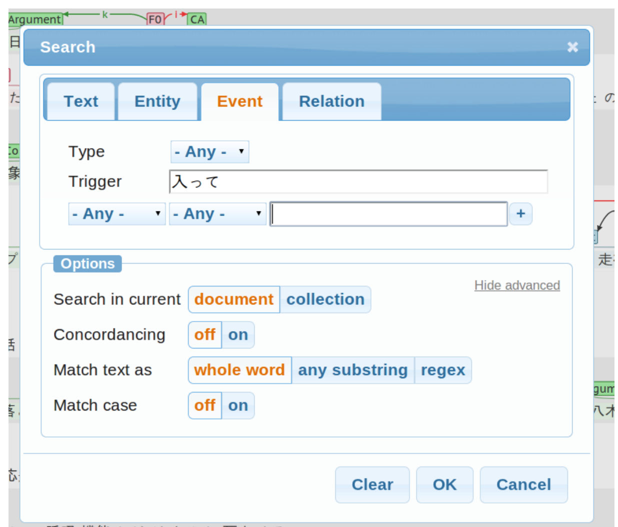

Figure 4: Screenshot of the STAV search-interface that enables searching for annotations based on their structure and relation to the underlying text

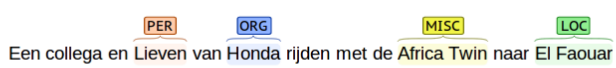

Figure 5: Named entity annotations

as search for all forms of annotation. Further, we enabled the user to place detailed constraints on the annotation structure targeted by the search (Figure 4). We additionally support keyword-in-context concordancing in the display of search results which gives the user a better view of which search results that may be relevant.

3.3 Conversion Tools

In order to demonstrate the applicability of our system to visualise a multitude of tasks, we constructed several conversion tools to convert popular formats for tasks such as dependency parses (Figure 1) and NER (Figure 5). It is our hope that this lowers the

[2] http://mecab.sourceforge.net/

barrier of entry to using the system since this enables a potential user to side-step the conversion step from established representations for many tasks.

3.4 New Annotation Primitives

As discussed in the introduction, STAV has been extended to support binary relations between entity annotation, a fundamental category of representation that extends the capabilities of the system to support a diversity of annotation tasks ranging from dependency syntax to many information extraction (IE) tasks.

Further, IE resources and tools are increasingly incorporating attributes such as confidence levels, intent, etc. for the annotated structures. Unlike simple attributes such as NEGATION, these annotations are not binary in nature, but can take multiple values (e.g. confidence level 1–5). Such attributes are now supported by STAV. This allows the visualisation of corpora annotated for e.g. meta-knowledge annotations [9], which further extends the set of compatible annotations.

4 Use-cases

This section discusses use-cases suggested by the STAV user-base, as well as examples of potential applications where the tool may prove useful.

4.1 End-user Visualisation

When presenting the results of annotation efforts and IE systems to end-users and professionals who are not familiar with the text domain or representation format, it is essential to find a visualisation that is understandable even to non-technical users.

One such application is utilising STAV to visualise the large EVEX [10] corpus that covers nearly 20,000,000 scientific abstracts to which a state-of-the-art IE system has been applied to extract event structure. By doing this, an end-user can now observe the location and nature of the extracted structures in the context of the text from which they were originally extracted. This effectively bridges the gap between IE researchers and end-users who wish to utilise the output of IE systems.

4.2 Tool for Error Analysis and Corpus Curation

An essential step of modelling natural language is to perform error analysis of a current model. This can be done either by manually stepping through a subset or all the annotations created by an automated system or by comparing the annotations produced

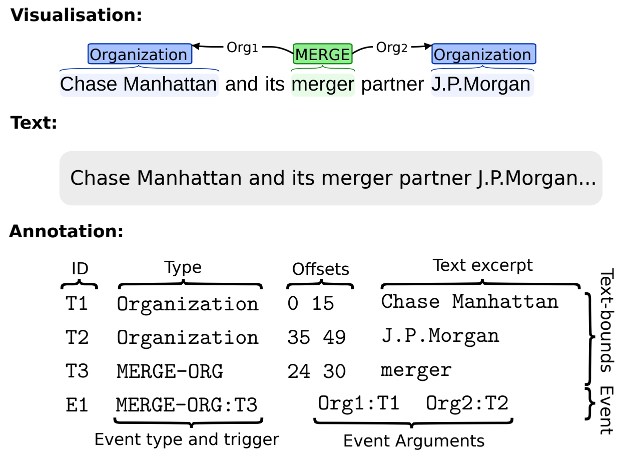

Figure 6: Illustration of how a simple event structure annotation relates to the four lines of stand-off format annotations which expresses them

by the system to gold annotations produced by human annotators. For both of these purposes STAV provides ways to view and compare different models and approaches in relation to each other and the gold annotations. This enables better and faster error analysis than viewing raw mark-up and eliminates the need to construct a complex visualisation system as a part of a particular research target and thus makes it possible to correct and observe differences in model behaviour at an early stage.

The case of disagreeing annotation sets is analogous to situations which frequently arise when constructing manual corpora with multiple annotators. Annotators may disagree upon certain constructions which are important to spot and reconcile at an early stage when producing high-quality corpora. For this purpose STAV can assist corpus curation due to its ability to search for problematic annotations to discuss with the annotators to reach a higher degree of inter annotator agreement.

5 Data Format

The data format used to represent the annotations visualised by STAV is a stand-off format derived from the format used for the BioNLP 2009 and 2011 Shared Tasks [5, 6]. Extensions to the format have been made to support aspects such as multi-valued attributes and notes, but the tool remains backwards compatible with any resource provided in the original format.

Figure 6 illustrates the structure of the data format using an example event structure based on the Automatic Content Extraction (ACE) event guidelines [7]. This text-based standoff format is straightforward to generate using automatic annotation tools and can be easily converted into from existing corpus annotations for visualizing them. As the format is line-oriented, it is further simple to process using standard *nix tools (`cut`, `grep`, `sort`, `uniq`, etc.) for search, analysis and statistics.

The basic primitives are the text-bound spans that are denoted by a leading T for their id;s. These text-bounds can then serve as triggers for N-ary event structures which can have restrictions on the types of text-bounds that may have certain roles. For relation and attribute annotations the targets of these annotations are also the text-bound annotations that serve as anchors and their syntax is similar, but for the sake of brevity they have been left out of this simple data format example.

6 Conclusions

In this work we have introduced the STAV text visualisation tool. A generalisation of an earlier tool constructed to handle event structures, STAV is now versatile enough to visualise a large variety of text-bound NLP annotations. We discussed new features such as extensive language support, search functionality, conversion tools, relations, and multi-valued attributes.

We discussed use-cases based on feedback from the NLP community following its initial release as a part of the BioNLP 2011 Shared task Among them how to use STAV for error analysis and to make IE results available to non-technical end-users.

It is our hope that STAV will continue to serve the community. The tool is freely available under an open-source license at: `http://github.com/nlplab/stav`

References

[1] Steven Abney. Parsing by chunks. *Principle-based parsing*, 44:257–278, 1991.

[2] Sabine Buchholz and Erwin Marsi. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City, June 2006. Association for Computational Linguistics.

[3] Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139 – 155, 2005.

[4] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 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, 2004.

[5] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[6] Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. Overview of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[7] LDC. ACE (Automatic Content Extraction) English Annotation Guidelines for Events. Technical report, Linguistic Data Consortium, 2005.

[8] Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. Bionlp shared task 2011: Supporting resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[9] Paul Thompson, Raheel Nawaz, John McNaught, and Sophia Ananiadou. Enriching a biomedical event corpus with meta-knowledge annotation. *BMC Bioinformatics*, 12(1):393, 2011.

[10] S. Van Landeghem, F. Ginter, Y. Van de Peer, and T. Salakoski. EVEX: a pubmed-scale resource for homology-based generalization of text mining predictions. In *Proceedings of BioNLP 2011 Workshop*, pages 28–37. Association for Computational Linguistics, 2011.

[11] N. Xue, F. Xia, F.D. Chiou, and M. Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(02):207–238, 2005.