Annotating Arguments: The NOMAD Collaborative Annotation Tool

Georgios Petasis
Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications
National Centre for Scientific Research (N.C.S.R.) "Demokritos"
GR-153 10, P.O. BOX 60228, Aghia Paraskevi, Athens, Greece
petasis@iit.demokritos.gr

Abstract

The huge amount of the available information in the Web creates the need for effective information extraction systems that are able to produce metadata that satisfy user's information needs. The development of such systems, in the majority of cases, depends on the availability of an appropriately annotated corpus in order to learn or evaluate extraction models. The production of such corpora can be significantly facilitated by annotation tools, which provide user-friendly facilities and enable annotators to annotate documents according to a predefined annotation schema. However, the construction of annotation tools that operate in a distributed environment is a challenging task: the majority of these tools are implemented as Web applications, having to cope with the capabilities offered by browsers. This paper describes the NOMAD collaborative annotation tool, which implements an alternative architecture: it remains a desktop application, fully exploiting the advantages of desktop applications, but provides collaborative annotation through the use of a centralised server for storing both the documents and their metadata, and instance messaging protocols for communicating events among all annotators. The annotation tool is implemented as a component of the Ellogon language engineering platform, exploiting its extensive annotation engine, its cross-platform abilities and its linguistic processing components, if such a need arises. Finally, the NOMAD annotation tool is distributed with an open source license, as part of the Ellogon platform.

Keywords: annotation tools, collaborative annotation, adaptable annotation schemas

1. Introduction

The development and maintenance of annotated corpora can be significantly facilitated through the use of annotation tools, as annotation tools can control most aspects of the annotation process, from the presentation of the relevant information to the annotators to the validation of annotated information according to a predefined schema. A plethora of annotation tools has been presented during the last decade (Uren et al., 2006; Fragkou et al., 2008), covering a wide range of annotation tasks and offering various levels of support. Annotation solutions can be divided into manual and semi-automatic methods: manual solutions provide the require infrastructure (i.e. storage management, graphical user interface, etc.) for annotators to annotate a corpus with a completely manual approach, where all information must be manually entered by the annotators. Semi-automatic solutions on the other hand, try to pre-annotate corpora, reducing the role of annotators into validation of existing pre-annotation. However, several of the existing annotation tools are desktop applications, allowing the annotation of corpora found on a single computer. A more recent category of annotation solutions, are *distributed* or *collaborative* annotation tools, where several annotators (not necessarily co-located) can annotate the same corpus, and in some cases even the same document. However, the construction of annotation tools that operate in a distributed environment is a challenging task, while the majority of these tools are implemented as Web applications (such as WebAnno[1] (Yimam et al., 2013) or BRAT[2] (Stenetorp et al., 2012)), having to cope with the capabilities offered by browsers. Annotator tools that operate as Web services are easier to implement, as the corpora are kept on a single server and annotation is happening also on the server, triggered by actions that happen at the browser of each annotator. Despite the fact the almost all collaborative annotation tools follow this approach, there are a few disadvantages in comparison to desktop applications:

- The graphical user interface of Web applications is less capable than a desktop user interface. Usability features like assigning keyboard-shortcuts to buttons, or special actions to mouse buttons, are usually missing.
- It is very difficult to personalise the annotation tool for each annotator. Features like monitoring the actions of a specific annotator, and inducing a set of regular expressions to pre-annotate documents, are quite difficult to be implemented.
- The annotation tool cannot store files or run application on the computer of the annotator. This suggests that any pre-annotation can only occur at the server, and not on the clients.
- An internet connection to the server is constantly required. An annotator cannot annotate locally, uploading the annotation results at a latter time.

This paper describes the NOMAD[3] collaborative annotation tool, which implements an alternative architecture: it remains a desktop application, fully exploiting the advantages of desktop applications, but provides collaborative annotation through the use of a centralised server for storing both the documents and their metadata, while exploiting instance messaging protocols for communicating events among all annotators. The annotation tool is implemented

[1] http://code.google.com/p/webanno/
[2] http://brat.nlplab.org/index.html
[3] http://www.nomad-project.eu

as a component of the Ellogon language engineering platform (Petasis et al., 2002), exploiting its extensive annotation engine, its cross-platform abilities and its linguistic processing components, if such a need arises. Finally, the NOMAD annotation tool is distributed with an open source license[4], as part of the Ellogon platform.

2. Related Work

A plethora of annotation tools has been made available to the NLP community during the last decade (Uren et al., 2006; Fragkou et al., 2008), targeting all related modalities (text, HTML, audio, video, etc.). Popular annotation tools like the ones included in GATE[5] (Cunningham et al., 2011), Ellogon (Petasis et al., 2002), the KIM Semantic Annotation Platform (Popov et al., 2004), the SHOE Knowledge Annotator (Heflin et al., 1999), Callisto[6], Wordfreak[7], MMAX2[8] (Müller and Strube, 2006), Knowtator[9] (Ogren, 2006), and AeroSWARM[10] (Corcho, 2006), allow the annotation of texts and HTML documents using either XML-based annotation schemas, or ontologies. Usually related with natural language engineering platforms/infrastructures, these tools are desktop applications that annotate corpora stored locally, on the same machine the annotation tool is used. On the other hand, there are several tools that allow the annotation of any Web page, such as A.nnotate[11], Bounce[12], Diigo[13], iComment[14], MyStickies[15], AnnotateIt [16] etc. Typically, these tools employ extensions that run inside a browser (usually developed in JavaScript) along with a centralised server (for storing the annotations), in order to allow the annotation of text and images in online material, such as Web pages. Usually, these annotations are free-form text fields, where users can type anything they wish. Not conforming to any annotation schema, required for structured annotation, these tools are not well suited to the same annotation tasks, as the tools aiming at linguistic annotation[17]. However, they offer some interesting advantages, including ease of use by not requiring installation of complex applications, the accurate rendering of HTML documents, and of course the possibility of distributed/collaborative annotation. Distributed/collaborative annotation tools not only offer the ability to create annotated corpora by annotators that are not co-located, but also offer the possibility of appealing to larger crowds, like the "Phrase Detectives" system[18] (Chamberlain et al., 2008), where linguistic annotation is exposed as a online game.

As a result, unifying the two categories of annotation tools is an appealing research area. Among the first approaches that tried to support distributed/collaborative annotation is the AGTK toolkit (Maeda and Strassel, 2004), which utilises a relational database for storing and accessing corpora on a shared server, in order to offer a framework for development of collaborative annotation tools. A similar approach is GATE Teamware (Bontcheva et al., 2010): utilising also a shared server, it offers an annotation tool that can be embedded through Java inside a browser. One of the main advantages of GATE Teamware is its extensive support for "roles", by separating annotators into three groups (managers, editors, annotators), and arranging their actions into annotation workflows. Similarly, WebAnno (Yimam et al., 2013) offers annotation project management, including the management of users in different roles. Based on technology from the BRAT rapid annotation tool (Stenetorp et al., 2012) for visualizing and editing annotations in a web browser, WebAnno allows POS, Named Entity, Dependency Parsing and co-reference resolution annotations, while offering extensive support for monitoring the overall annotation process, detecting contradictions among annotators and calculating inter-annotation agreement. The NOMAD annotation tool shares architectural elements from both systems, as:

- Utilises a central server for storing corpora, similar to AGTK, Teamware and WebAnno. However, the server needs to store only a relational database, as in AGTK, avoiding the complex installation process of Teamware and WebAnno[19], or the need to use a commercial hosting platform like GateCloud[20].
- The NOMAD annotation tool from the annotator perspective is a desktop application, distributed as a single executable, similarly to AGTK. However, the NOMAD tool is adaptable to XML annotation schemas, similar to the Teamware tool.
- Supports management of corpora, allowing the creation/deletion of collections, and the addition/deletion of documents into them.

In addition, the NOMAD annotation tool introduces several novel aspects. Unique features of the NOMAD annotation tool with respect to the state of the art, include:

- Data integrity: All data held in the central server are also stored also locally, by every instance of the NOMAD annotation tool. This ensures that data and operation can be immediately restored in case of a problem in the central server, or in case a new server acquires

[4]The NOMAD annotation tool is distributed under the LGPL version 3 license. Available from: http://www.ellogon.org/index.php/annotation-tool/nomad-annotation-tool

[5]http://gate.ac.uk/

[6]http://callisto.mitre.org/

[7]http://wordfreak.sourceforge.net/

[8]http://www.eml-research.de/english/research/nlp/download/mmax.php

[9]http://bionlp.sourceforge.net/Knowtator/index.shtml

[10]http://projects.semwebcentral.org/projects/aeroswarm/

[11]http://a.nnotate.com/

[12]http://www.bounceapp.com/

[13]http://www.diigo.com/

[14]http://www.icomment.com/

[15]http://www.mystickies.com/

[16]http://annotateit.org/

[17]AnnotateIt is open source and extensible through plugins developed in JavaScript, allowing developers to organise annotation into fields, resembling structured annotation

[18]http://anawiki.essex.ac.uk/phrasedetectives/instructions.php

[19]WebAnno offers also WebAnnoStandalone, which simplifies installation.

[20]https://gatecloud.net/

the role of the central repository. Multiple copies of the data ensure that the corpora will never be lost, even if something happens to the central server.

- Off-line annotation mode: An annotator can lock a document, annotate it without being connected to the central server, and upload modifications at a later time.
- Robustness: Communication with the central server is not required during document annotation, making the annotation process immune to network connection temporal errors/drops.
- Personalisation of the tool to the needs of each annotator: the NOMAD annotation tool allows the assignment of keyboard shortcuts/mouse shortcuts that are unique for each annotator, and stored locally by every tool instance.
- Data export: ability to export any corpus/collection/document in various formats, including XML, directly from the tool.
- Personalised automatic annotation support, through regular expression acquisition performed by monitoring the actions of each annotator.
- Easy to setup and administer, as the tool is distributed as a single executable file, and the central repository can be filled automatically by any instance of the tool, if it is new or empty.

In addition, the NOMAD tool includes a complete distribution of the Ellogon language engineering platform, suggesting that all its processing components are available from inside the tool, and can be applied on a document/collection. Teamware offers similar functionality, but execution of linguistic processing components can happen only at the server, and not within the tool. Local execution of components within the NOMAD tool allows access to the vast collection of Ellogon's components, including operators that transform imported documents, linguistic annotation viewers, annotation schema validators, inter-annotation agreement calculators, etc.

3. Annotating Arguments

Argumentation is a branch of philosophy that studies the act or process of forming reasons and of drawing conclusions in the context of a discussion, dialogue, or conversation. Being an important element of human communication, its use is very frequent in texts, as a means to convey meaning to the reader. As a result, argumentation has attracted significant research focus from many disciplines, ranging from philosophy to artificial intelligence. Central to argumentation is the notion of argument, which according to (Besnard and Hunter, 2008) is "a set of assumptions (i.e. information from which conclusions can be drawn), together with a conclusion that can be obtained by one or more reasoning steps (i.e. steps of deduction)". The conclusion of the argument is often called the claim, or equivalently the consequent or the conclusion of the argument, while the assumptions are called the support, or equivalently the premises of the argument, which provide the reason (or equivalently the justification) for the claim of the argument. The process of extracting conclusions/claims along with their supporting premises, both of which compose an argument, is known as argument extraction (Goudas et al., 2014) and constitutes an emerging research field.

NOMAD is an EU-funded project that aims to aid modern politicians in testing, detecting and understanding how citizens perceive their own political agendas, and also in stimulating the emergence of discussions and contributions on the informal web (e.g. forums, social networks, blogs, newsgroups and wikis), so as to gather useful feedback for immediate (re)action. In this way, politicians can create a stable feedback loop between information gathered on the Web and the definition of their political agendas based on this contribution. The ability to leverage the vast amount of user-generated content for supporting governments in their political decisions requires new ICT tools that will be able to analyze and classify the opinions expressed on the informal Web, or stimulate responses, as well as to put data from sources as diverse as blogs, online opinion polls and government reports to an effective use. NOMAD aims to introduce these different new dimensions into the experience of policy making by providing decision-makers with fully automated solutions for content search, selection, acquisition, categorization and visualization that work in a collaborative form in the policy-making arena.

One of the central elements within the NOMAD project is the identification of *arguments* in favour or against a topic, and the opinion polarity expressed on the informal Web towards these arguments. For the purposes of evaluation of the NOMAD system, a "gold" manually annotated corpus has been created for all three languages involved in the project (German, English, Greek), containing 500 documents gathered from the Web for each language, relevant to the thematic domains of the project (open data, allergies and renewable energy sources). A suitable annotation schema has been devised for the task, which allows the annotators to define a hierarchy of arguments from arguments they identify in texts, mark the segments that represent the argument (including the claim and premise, along with entities related to the argument) and associate a polarity to the argument, representing the opinion polarity of the document author towards this argument.

4. The NOMAD Annotation Tool

In order to develop an annotation tool that would be easy to use and generic enough to support a wide range of annotation tasks, we identified four basic requirements for our system: *a*) The tool should be user-friendly, easy to be understand and operated by the annotators. *b*) The operation must be based on annotation schemas that define the annotation task and guide the annotators in their work. The annotation tool must adapt its user interface automatically according to the loaded annotation schema. *c*) The system should support collaborative/distributed annotation, where the annotation process can be shared among different annotators at different locations. *d*) The system should be tolerant to losses of internet connectivity, allowing the annotation to continue locally, if possible. The architecture of

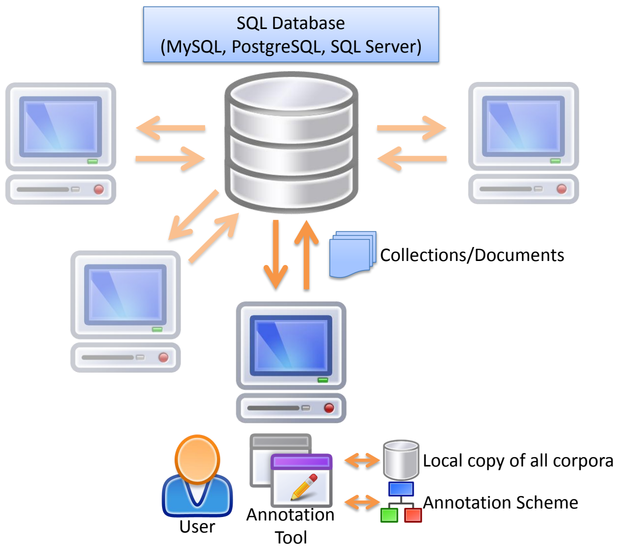

Figure 1: The architecture of the NOMAD annotation tool.

the NOMAD annotation tool is shown in figure 1. The central component is an SQL database server, where all tools are registering themselves upon start-up and termination. The database server is used to store collections of documents, either annotated or not, along with any other information required by the annotation tools. Each annotation tool communicates with the database server through SQL queries, and the supported databases are MySQL[21], PostgreSQL[22], and Microsoft SQL Server[23]. A single instance of the NOMAD annotation tool can be run in any number of computers. Each instance registers itself with the database server, using the credentials of the user running the application (taken from the operating system), and synchronises its local copy with the database server. Each annotation tool stores locally all information kept in the server, ensuring that all data can be restored even if the database server gets replaced with a new server, and at the same time providing the ability to annotate off-line.
The main window of the annotation tool is shown in figure 2, while the window for annotating documents (with the NOMAD annotation schema loaded) is shown in figure 3. The main window is organised around collection and document management, where any annotator can create/delete/modify collections by adding or removing documents. The annotation window adapts itself to the selected annotation schema, allowing the user to select segments and annotate them with a set of attributes, according to the annotation schema. The user has complete control over the way segments are selected (i.e. by configuring mouse buttons or key combinations to select whole words), and annotated (i.e. by configuring key combinations for any category). In addition, the tool monitors the annotation performed by the user, and tries to extract regular expressions

[21]http://www.mysql.com/
[22]http://www.postgresql.org/
[23]http://www.microsoft.com/sqlserver/en/us/default.aspx

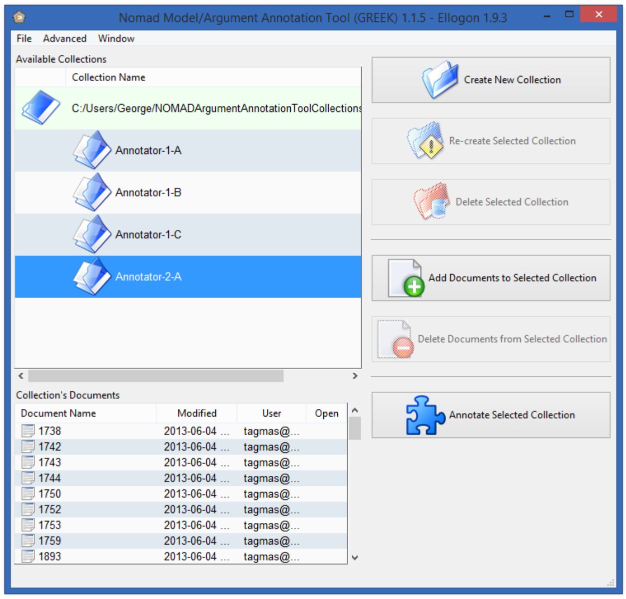

Figure 2: The main window the NOMAD annotation tool.

from already annotated items. The user is able to revise these expressions (if desired) and apply them to automatically annotate the rest of the document, or other documents. The NOMAD tool supports both distributed and collaborative annotation. The distributed mode is the most frequent and easy to use mode: each annotator locks a whole document for editing, by simply opening this document in its tool, preventing any other annotator to open the same document for annotation. The collaborative mode is more complex, and requires a different configuration of the tool. Withe the help of the open instance messaging protocol XMPP[24], also used by Google Talk, actions performed by annotators are shared among all instances of the tool, effectively sharing annotations among all users that annotate the same document. However, no conflict resolution is performed: if two users annotate the same text segment, both annotations are kept into the system, no matter if they are overlapping or contradicting. Finally, the annotation tool currently runs under the Windows (XP, Vista, 7) and Linux (32 and 64 bit) operating systems.

5. Reusing Ellogon's Annotation Engine

The Ellogon language engineering platform (Petasis et al., 2002) offers an extensive annotation engine, allowing the construction of a wide range of annotation tools for both plain text and HTML documents. This annotation engine provides a wide range of features, including: *a*) cross-platform graphical user interface (supporting Windows, Linux and OS X), *b*) use of standard formats (including stand-off annotation in XML), *c*) support for user centered design and user friendly interface, *d*) support of customized annotation schemata, *e*) support for annotating rendered HTML pages, *f*) support for performing automatic annotation, and *g*) comparison facilities, to identify mismatches

[24]The Extensible Messaging and Presence Protocol (XMPP): http://www.jabber.org/

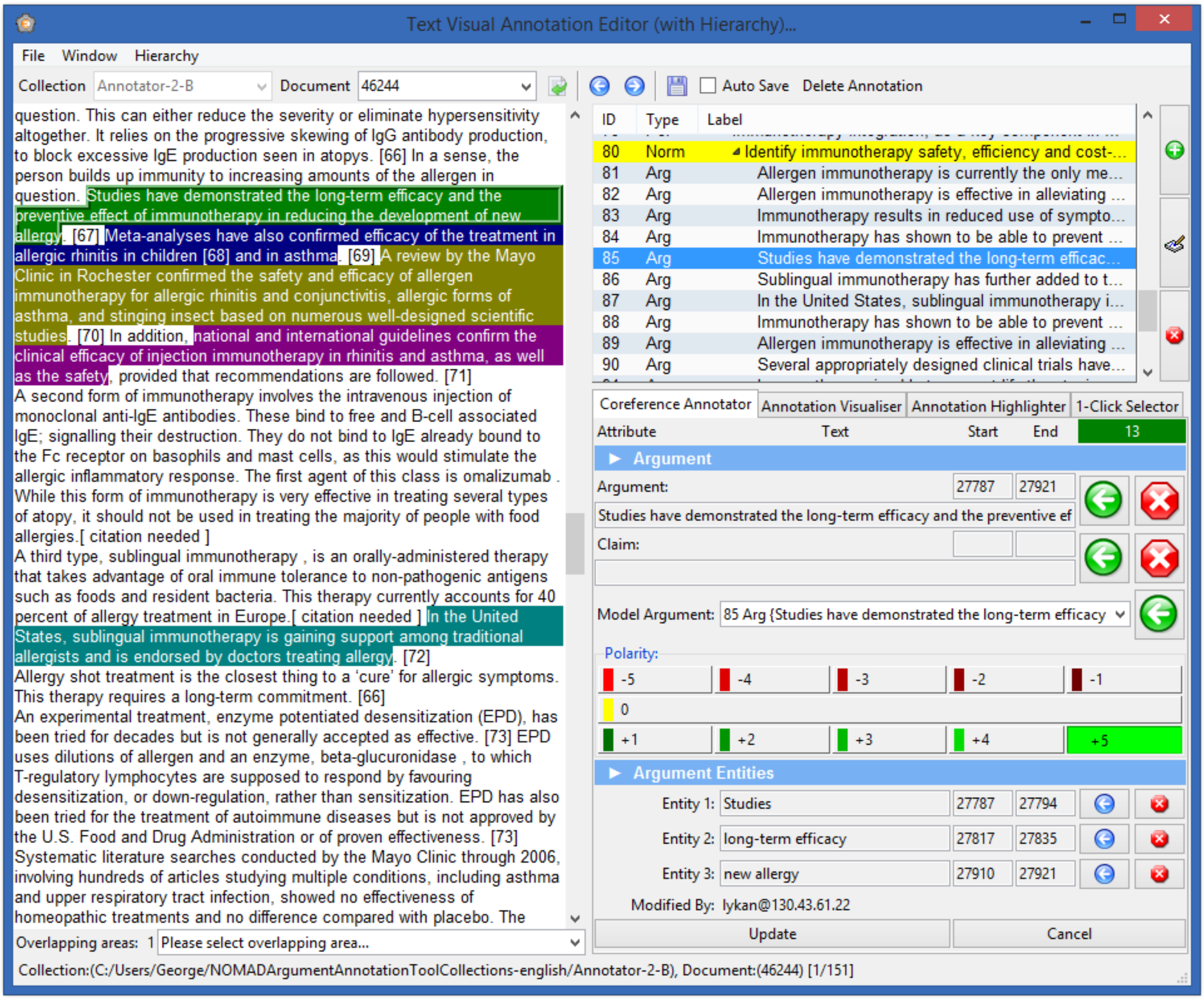

Figure 3: The annotation window of the NOMAD annotation tool.

among independent annotations of the same document, or calculate inter-annotation agreement. Despite the fact that these features are not unique among the available annotation tools (i.e. most of these features are also supported by tools offered by Callisto, Wordfreak, GATE, MMAX2, Knowtator, and AeroSWARM), reusing an annotation engine allows for rapid and robust development of a new annotation tool, through the re-use of tested components.

5.1. Annotation Schemas

The annotation engine of the Ellogon language engineering platform is configurable through XML files, that define annotation schemas. The tool reads the annotation schema from an XML file, and presents to the annotator a suitable GUI for annotating text segments. The NOMAD tool follows a different approach than other tools, such as GATE Teamware: Instead of implementing floating windows which show only a small fragment of the annotation schema, the NOMAD tool shows the whole schema, so as not to impose to the user the need to perform excessive mouse usage. In addition, the colours are not related to annotation groups (as in GATE Teamware), but on category/attribute values. The XML annotation schema language provides a variety of types that can be annotated. The most important types, along with their visual representation in the GUI, are shown in the following list:

- A **category** (figure 4-A) can be used to assign a specific category to selected segments. It is usually represented by a button widget. Typical usage of this schema type is to annotate POS tags, named-entity types, polarity, etc.

- A **date** (figure 4-B) can be used to assign a specific category to selected segments and in addition associate a date. It is usually represented by a button widget along with a date picker, to select the associated date. The date can be formatted according to the format specified in the annotation schema. Typical usage of this type is when dates in text must be associated/grounded with a normalised date, such as marking the text segment "yesterday", and ground its date to "21 May 2012" in the linguistic annotation.

- A **category** with a **description** (figure 4-C). This in-

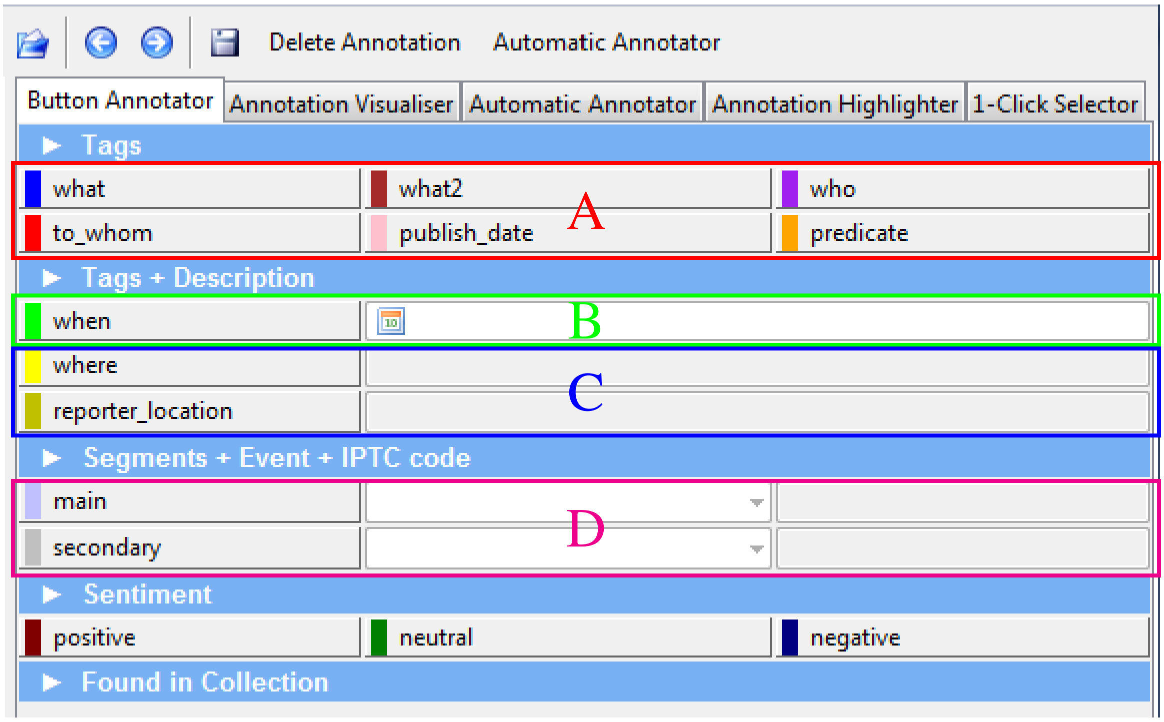

Figure 4: Various types of annotation input.

put type can be used to assign a category and an arbitrary description (comment) to a selected segment. It is usually represented by a button widget and an entry widget, allowing the entry of arbitrary text. This type can be used when a category may be associated with a note/comment from the annotator.

- A **category** with a **detail** and possibly a **description** (figure 4-D). This input type can associate a category to a segment, along with a "detail", a sub-category of predefined values, along with a description (arbitrary text). It is usually represented by a button widget and a combo-box widget, allowing the selection of a value among a set of predefined values. In case an optional description has been specified in the annotation schema, it is represented by an entry widget, allowing the entry of arbitrary text. Typical usage of this type is when a category has too many values to be represented as buttons, and the values should be selected from a drop-down list. For example, the NOMAD annotation schema (shown in figure 3) includes a widget for defining "events" in its top-right corner: the annotator may identify the events contained in a news item, and define them, with a small description. Each defined event gets a unique id. In addition to the event definition, the annotator can mark the segment from where the event was extracted (figure 4-D), and select from the drop-down list the id of the defined event, so as to associate a segment to an event definition, which may be the same across many documents.

In addition, there are some types of annotation input that relate to grouping several segments and other information in a single annotation, to facilitate annotation of co-reference or other types of relations:

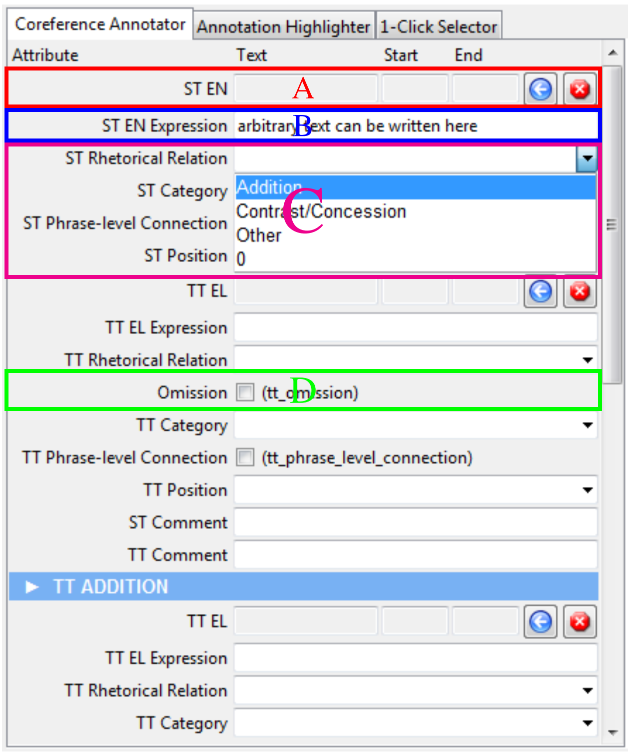

Figure 5: Types of annotation input within groups.

- A **span** or **segment** (figure 5-A). This input type is represented by a textual label (specified by the annotation schema), the text of the segment, its offsets, a button to fill in the segment from the current selection, and a button to clear the segment. It should be noted that the annotator is not required to type anything. For example, if the annotation schema defines a “source” attribute, the annotator is expected to either select an already annotated segment, or select a text segment with the mouse, and press the button with the blue arrow icon, to fill the “source” property.

- A **description** (figure 5-B), which the user can fill with arbitrary text. Represented by a textual label and an entry widget, where arbitrary text can be entered.
- A **category** (figure 5-C), selectable from a set of predefined categories by the annotation schema. Represented by a textual label and a combo-box widget, allowing the user to select a category from a set of predefined categories.
- A **boolean value** (figure 5-D), denoting the presence or absence of an attribute. Represented by a textual label and a check-box widget.

Finally, the annotation inputs can be separated in groups having a label, through the annotation schema, as shown in figure 4. Some more annotation input types related to template element filling, can be found in (Fragkou et al., 2008) and (Petasis, 2012), while an annotation schema for linguistic analysis of connectives on bi-lingual aligned corpora, can be found in (Tsoumari and Petasis, 2011).

5.2. The NOMAD Annotation Scheme

The Annotation Tool has been used in the context of the NOMAD project, in order to create a corpus, manually annotated with arguments (claims and premises), along with sentiment information capturing the author’s opinion towards each argument. Annotators had to perform two main activities: *a*) Identify arguments: Each annotator had to read the text, and identify any argument related to the domain of interest (renewable energy sources for the Greek sub-corpus, allergy and immunotherapy for the English sub-corpus, and op[en data for the German sub-corpus). Each argument had to be added into a hierachy of policies, sub-policies, and arguments in favour or against specific sub-policies (shown in the upper-right part of figure 3). *b*) Annotate arguments and sentiment in text: Each annotator must mark text segments that represent to argument components (claim and premises), associate these segments with an argument from the hierarchy, associate the argument with a set of named entities (specific to the thematic domain), and mark the snetiment of the document author towards the specific argument, as identified from the text passage.

As shown in figure 3, the annotation window is split into two large, vertical areas. On the left side of the annotation window, a rendering of the document is provided, where the annotator can read the document and mark segments to be annotated. The right side of the annotation window provides the annotation facilities as described by the annotation scheme. It is spitted vertically into two major parts: The upper part displays the policy model (in the form of a hierarchy), which the annotator can freely manage: She/he can add new nodes (at any level), move them around with drag and drop, edit them, or delete them. This hierarchy acts as a simplistic model of one or more policies. Policies are what a policy maker wants to achieve, and each policy can be decomposed into policy components: subpolicies, representing ways to achieve a policy, and arguments in fovour or against a sub-policy. The lower part of the right column shows the fields that relate to the argument annotation, where the annotator is expected to fill the required fields or slots with text segments from the document. It consists primarily from a set of spans or text segments (fields “Argument”, “Claim”, “Entity 1-3”), the policy argument (from the hierarchy) to associate with the annotation (“Model Argument”), and the author’s sentiment (Polarity), accepting a value from -5 (very negative) to +5 (very positive).

6. Conclusions and Future Work

In this paper we presented a new distributed/collaborative annotation tool, which tries to combine distributed/collaborative annotation with desktop applications, following a different approach from Web based distributed annotation tools. The presented annotation tool is implemented as an extension (plug-in) of the Ellogon language engineering platform, exploiting facilities like graphical user interface elements and its extensive annotation engine. The annotation tool has been used in the context of the NOMAD research project, in order to annotate arguments on documents retrieved from the Web (including social media), and polarity towards these arguments. As future work, we aim to enhance the ability of annotating a single document by more than one annotator, especially towards conflict resolution, as the currently provided conflict resolution facilities are quite limited. In addition, we are examining the possibility to develop a Web-based counterpart, providing only the functionality that can be ported in a Web-based UI, in the context of the CLARIN-EL project.

Acknowledgments

The authors would like to acknowledge partial support of this work from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288513. For more details, please see the NOMAD project’s website, `http://www.nomad-project.eu`.

7. References

Besnard, P. and Hunter, A. (2008). *Elements of Argumentation*. MIT Press.

Bontcheva, K., Cunningham, H., Roberts, I., and Tablan, V. (2010). Web-based collaborative corpus annotation: Requirements and a framework implementation. In Witte, R., Cunningham, H., Patrick, J., Beisswanger, E., Buyko, E., Hahn, U., Verspoor, K., and Coden, A. R., editors, *New Challenges for NLP Frameworks (NLPFrameworks 2010), in Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*, Valletta, Malta, May 22. ELRA.

Chamberlain, J., Poesio, M., and Kruschwitz, U. (2008). Phrase detectives - a web-based collaborative annotation game. In *In Proceedings of I-Semantics*.

Corcho, O. (2006). Ontology based document annotation: trends and open research problems. *Int. J. Metadata Semant. Ontologies*, 1:47–57, January.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*.

Fragkou, P., Petasis, G., Theodorakos, A., Karkaletsis, V., and Spyropoulos, C. (2008). Boemie ontology-based text annotation tool. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Goudas, T., Louizos, C., Petasis, G., and Karkaletsis, V. (2014). Argument extraction from news, blogs, and social media. In *A. Likas and K. Blekas and D. Kalles (Eds.): SETN 2014, LNCS 8445*, pages 287–299. Springer International Publishing Switzerland.

Heflin, J., Hendler, J., and Luke, S. (1999). Shoe: A knowledge representation language for internet applications. Technical report.

Maeda, K. and Strassel, S. (2004). Annotation tools for large-scale corpus development: Using AGTK at the Linguistic Data Consortium. *Proceedings of the* 4^{th} *International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal*.

Müller, C. and Strube, M. (2006). Multi-level annotation of linguistic data with MMAX2. In Braun, S., Kohn, K., and Mukherjee, J., editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany.

Ogren, P. V. (2006). Knowtator: A protégé plug-in for annotated corpus construction. In Moore, R. C., Bilmes, J. A., Chu-Carroll, J., and Sanderson, M., editors, *HLT-NAACL*. The Association for Computational Linguistics.

Petasis, G., Karkaletsis, V., Paliouras, G., Androutsopoulos, I., and Spyropoulos, C. D. (2002). Ellogon: A New Text Engineering Platform. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 72–78, Las Palmas, Canary Islands, Spain, May 29–31. European Language Resources Association.

Petasis, G. (2012). The SYNC3 Collaborative Annotation Tool. In *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, pages 363–370, Istanbul, Turkey, May. European Language Resources Association.

Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., and Kirilov, A. (2004). KIM - a semantic platform for information extraction and retrieval. *Journal of Natural Language Engineering*, 10(3-4):375–392.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). Brat: A web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 102–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tsoumari, M. and Petasis, G. (2011). Coreference Annotator - A new annotation tool for aligned bilingual corpora. In *Proceedings of the Second Workshop on Annotation and Exploitation of Parallel Corpora (AEPC 2), in 8th International Conference on Recent Advances in Natural Language Processing (RANLP 2011)*, pages 43–52, September 15.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., and Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semant.*, 4:14–28, January.

Yimam, S. M., Gurevych, I., de Castilho, R. E., and Biemann, C. (2013). Webanno: A flexible,web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 1–6, Stroudsburg, PA, USA, August. Association for Computational Linguistics.