Djangology: A Light-weight Web-based Tool for Distributed Collaborative Text Annotation

Emilia Apostolova, Sean Neilan, Gary An*, Noriko Tomuro, Steven Lytinen

College of Computing and Digital Media, DePaul University,Chicago, IL 60604 U.S.A.
*Department of Surgery, Northwestern University Feinberg School of Medicine, Chicago, IL 60611 U.S.A.
emilia.aposto@gmail.com, sean@seanneilan.com, docgca@gmail.com,
tomuro@cs.depaul.edu, lytinen@cs.depaul.edu

Abstract

Manual text annotation is a resource-consuming endeavor necessary for NLP systems when they target new tasks or domains for which there are no existing annotated corpora. Distributing the annotation work across multiple contributors is a natural solution to reduce and manage the effort required. Although there are a few publicly available tools which support distributed collaborative text annotation, most of them have complex user interfaces and require a significant amount of involvement from the annotators/contributors as well as the project developers and administrators. We present a light-weight web application for highly distributed annotation projects - Djangology. The application takes advantage of the recent advances in web framework architecture that allow rapid development and deployment of web applications thus minimizing development time for customization. The application's web-based interface gives project administrators the ability to easily upload data, define project schemas, assign annotators, monitor progress, and review inter-annotator agreement statistics. The intuitive web-based user interface encourages annotator participation as contributors are not burdened by tool manuals, local installation, or configuration. The system has achieved a user response rate of 70% in two annotation projects involving more than 250 medical experts from various geographic locations.

1. Introduction

Text annotation is an inherent step in almost any Natural Language Processing (NLP) task. Ever since the Message Understanding Conference (MUC) was first held in 1987, the creation of annotated corpora and evaluation metrics has been a focus in the NLP community. Sharable common resources such as gold standards and training corpora related to various NLP tasks have been actively created for over 20 years. The ISO/TC 37/SC4 Language Resources Management sub-committee [1] of the International Organization for Standardization (ISO) is actively developing annotation standards (Ide and Romary, 2004). Even though the need for annotation standards has been recognized, historically annotated corpora and NLP frameworks employ ad-hoc annotation schemas. As a result, the task of the NLP developer/researcher inevitably involves ad-hoc translation from one annotation coding scheme to another.

The repository of existing publicly available annotated corpora has been growing (the catalogue of the Linguistic Data Consortium currently consists of 450 linguistic corpora[2]). However, most real-world NLP efforts tackle new domains, languages, or tasks and lack the support and convenience of pre-existing annotated data. Manual annotation of textual data is known to be a time- and resource-consuming task. The task is further complicated by expectations of the involvement of multiple annotators for the purpose of developing objective metrics based on inter-annotator agreement statistics. To address this problem, an idea of distributing the effort across multiple groups/contributers has emerged. This paper presents a light-weight web application for collaborative, distributed annotation of text documents - Djangology. This web application provides easy integration with existing annotation schemas, as well as a basis for rapid development of customized web-based annotation tools and information models.

2. Related Work

It is not uncommon for NLP research groups to rely on manual text annotation tools which are developed in-house and for specific domains or tasks. However, the use of general-purpose text annotation tools (providing various degrees of customization capabilities) has become more common recently. A number of stand-alone annotation applications have been made publicly available to the NLP research community. Callisto (Day et al., 2004) is a Java annotation framework which provides a plug-in development environment for custom annotations. MMAX2 (Müller and Strube, 2006) and Word-Freak (Morton and LaCivita, 2003) are also Java-based stand-alone tools which support annotation schema definitions and custom XML-based annotation exports. The GATE NLP framework (Cunningham et al., 2002) integrates an annotation interface to the framework's Java GUI (Graphical User Interface). A couple of annotation plug-ins have been developed for the Protégé [3] Java framework - iAnnotateTab (Chintan, 2005) and Knowtator (Ogren, 2006). The Knowtator plug-in has gained popularity and allows complex annotation schema design, inter-annotator statistics, as well as the creation of a gold standard based on annotations from multiple annotators.

As stand-alone applications, all of the annotation tools above require significant involvement from the annotators - software installation and typically non-trivial configuration. Accumulating, evaluating, and consolidating annotations from various annotators also involves considerable effort. Web-based distributed annotation tools have been de-

[1] http://www.tc37sc4.org/index.php
[2] http://www.ldc.upenn.edu/Catalog/
[3] http://protege.stanford.edu

veloped to avoid the hassle of stand-alone applications and to streamline collaborative annotation efforts. The GATE framework (Cunningham et al., 2002) supports collaborative annotation through the OLLIE client-server application (Cunningham et al., 2003). The OLLIE client uses a Java-enabled web browser and communicates to a server via Java Applets. The web-based application uses GATE's distributed database facilities to authenticate and store user data as well as the language resources being edited. One drawback of the system is that communication between the web-based client and the server is achieved via Java RMI (Remote Method Invocation), which has limited browser support and is known to have a number of deployment difficulties that render the approach impractical. Serengeti (Stührenberg et al., 2007) is a project-specific Mozilla Firefox plug-in that allows web-based annotation of anaphoric relations using a pre-defined schema. Annozilla [4] is another Mozilla Firefox plug-in that allows spans of text in html/text documents to be associated with annotation types and free text. Annotations can be stored in a RDF-format local datastore on the user's machine, or posted/retrieved from a compliant Semantic Web annotation server.

3. Motivation and Case Study

The Djangology[5] annotation web application was originally created to meet the needs of a collaborative annotation project involving more than 250 international participants. The goal of the project was to create a gold standard corpus which is annotated with named entities of the domain of interest: medical studies of trauma, shock, and sepsis conditions. Abstracts from an annual conference dedicated to the subject and hosted by the the North American Shock Society [6] were used to identify the domain-specific named entities via an automated process. The named entity annotations had to then be validated by domain experts - the contributors to the conference. The Djangology system has been in use for two consecutive years (2008 and 2009), and has achieved an average contributor response rate of 70%.

The needs of the project led to a set of requirements common to similar highly-distributed collaborative annotation projects. An administration interface was needed to manage documents and users, as well as for the definition of annotation schemas. Annotations created via an automated process needed to be loaded into the system. Participants were notified via email and presented with a link to the web-based interface. After logging in, annotators were able to view a list of assigned documents. An intuitive web-based user interface was needed to allow participants to annotate documents with minimal instructional text. Easy and quick annotation access was crucial to the success of the project. As the time of domain experts is quite valuable, complicated installation or annotation instructions would be prohibitive. The system also needed to display inter-annotator agreement statistics, as well as the evaluation statistics comparing a gold standard against the automated annotation.

4. System Description

4.1. Technical Details

Recent years have introduced new technologies for rapid and easy development and deployment of web applications, most notably the Ruby on Rails[7] framework (developed for the Ruby language) and the comparable Django framework[8] (based on the Python language). For our annotation project, the Django web framework was selected because the framework has excellent documentation and design, which allows for the development of high-performing, elegant web applications quickly. In addition, the Python programming language has traditionally been in wider use in the academic community (compared to Ruby) and a number of Python NLP frameworks and tools could be easily integrated into the web application if necessary.

In terms of deployment, Django requires almost no configuration as it is based on the software design principle *Convention over Configuration*. It supports almost all popular database servers, including PostgreSQL, MySQL, Oracle and SQLite. An out-of-the-box administration web interface provides facilities for user and database record management, thereby reducing development time significantly. Modifying the database schema requires minimal effort on the developer through the use of the *Active Record* design pattern. Similarly, the framework provides support for rapid development of custom pages or modifying existing interfaces. In addition, Django supports *agile* development practices through built-in automated testing and support for rapid unit test writing.

Djangology can be deployed on any web-accessible server and requires a Python installation, Django installation, and connectivity to a database server [9]. Source code and installation instructions can be found at the project website http://djangology.sourceforge.net/. We estimate that end-to-end installation and configuration time for a Python and Django-savvy developer is less than an hour. Once deployed, the application can be accessed from any web browser - no browser plug-ins, JVM installation, or custom security settings are necessary, as the client-server communication is based on standard HTTP and Ajax requests.

The application database schema (Figure 1(a)) and user interface can be rapidly extended and customized. For example, creating a new field to annotator accounts could be effortlessly achieved by just adding a new attribute to the corresponding Python model class. The corresponding web form and underlying database schema are transparently updated by the Django framework (Figures 1(b) and 1(c)).

4.2. User Interface and Workflow

The Djangology [10] application presents administrators with an interface to create/modify annotation projects and manage users (Figure 2). Administrators can import documents

[4] http://annozilla.mozdev.org/

[5] The system is named after the 1949 Django Reinhardt album. The name also highlights the use of the Django web framework.

[6] http://www.shocksociety.org/

[7] http://rubyonrails.org/

[8] http://www.djangoproject.com/

[9] Quick install guide can be found at http://docs.djangoproject.com/en/dev/intro/install/

[10] http://sourceforge.net/projects/djangology/

(single document or batch mode) into a project, define the project annotation schema, create annotator accounts, and assign annotators to specific projects and to a list of documents. Existing annotations and documents could also be easily loaded into the system through custom Python scripts (stand-alone Django scripts) or through direct connection to the Djangology database. Djangology has been used to import manually created annotations in the Knowtator format and from the BioScope Corpus (Szarvas et al., 2008) as well as annotations created automatically by the Gate and UIMA (Ferrucci and Lally, 2004) frameworks and the Metamap © system from the National Library of Medicine.
In the workflow of the system, contributors are typically emailed their system authentication information and presented with a link to the application (Figure 3). Once logged in, annotators can select one of their assigned documents and proceed with the web-based annotation interface. An Ajax-based web page allows contributors to highlight a fragment of text and assign it to one of the pre-defined annotation types (based on the project annotation schema). The procedure for entering new annotations and modifying existing annotations is intuitive and based on user interface conventions - text selection/right-click menu selection. The system is specifically designed to require minimum time-investment on the part of the involved annotators. No installation, configuration, or reading user manuals is necessary on the part of the contributors. Annotations are saved to the backend database as they are entered, ensuring that no work is lost. In order to save annotators' effort, once a phrase is annotated, all occurrences of the phrase in the document are automatically annotated in the same type. Users are also given a facility to override the automatically created annotations or change the system's default behavior. If desired, contributors could also mark documents as completed to alert the project administrator of the annotation progress.
Once annotations are gathered from various contributors, project administrators have the ability to view inter-annotator agreement statistics - a variety of pair-wise project-based and document-based metrics are computed and presented in the user interface (Figure 4(a)). As analysis of inter-annotator disagreement is a common task, an interface for a side-by-side comparison of document annotations is also provided (Figure 4(b)).

5. Conclusions

This paper presented a light-weight open-source framework supporting distributed annotation projects. The framework was built to satisfy project needs currently not supported by existing publicly available annotation tools - minimal time commitment on the part of distributed contributors and minimal development and configuration effort on the part of project administrators. We estimated that installing both Django and Djangology takes less than one hour for a Python-savvy programmer. As contributors can access the annotation project and perform annotations online, no installation and configuration time-investment is required from annotators. An intuitive and standard user interface facilitates active project participation.
The Django web framework allows rapid development of extensions and customizations without limiting users to a pre-defined set of configuration files or requiring investment in complex, heavy-weight applications. By open-sourcing the framework, we hope to receive valuable feedback from the community and utilize it to prioritize the features which we are planning to incorporate in future releases.

6. References

P. Chintan. 2005. iAnnotate Tab.

D.H. Cunningham, D.D. Maynard, D.K. Bontcheva, and M.V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications.

H. Cunningham, V. Tablan, K. Bontcheva, and M. Dimitrov. 2003. Language engineering tools for collaborative corpus annotation. In *Proceedings of Corpus Linguistics*. Citeseer.

D. Day, C. McHenry, R. Kozierok, and L. Riek. 2004. Callisto: A configurable annotation workbench. In *International Conference on Language Resources and Evaluation*. Citeseer.

D. Ferrucci and A. Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

N. Ide and L. Romary. 2004. International standard for a linguistic annotation framework. *Natural language engineering*, 10(3-4):211–225.

T. Morton and J. LaCivita. 2003. Word-Freak: an open tool for linguistic annotation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations*, pages 17–18.

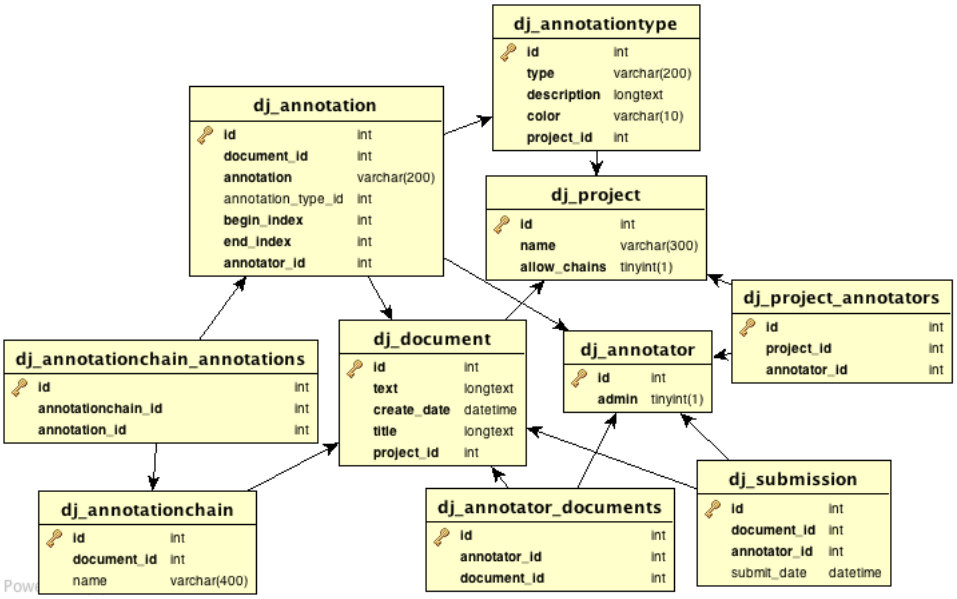
C. Müller and M. Strube. 2006. Multi-level annotation of linguistic data with MMAX2. *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods. Peter Lang, Frankfurt aM, Germany*.

P.V. Ogren. 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: companion volume: demonstrations*, page 275. Association for Computational Linguistics.

M. Stührenberg, D. Goecke, N. Diewald, A. Mehler, and I. Cramer. 2007. Web-based annotation of anaphoric relations and lexical chains. In *Proceedings of the ACL Linguistic Annotation Workshop*, pages 140–147.

G. Szarvas, V. Vincze, R. Farkas, and J. Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45. Association for Computational Linguistics.

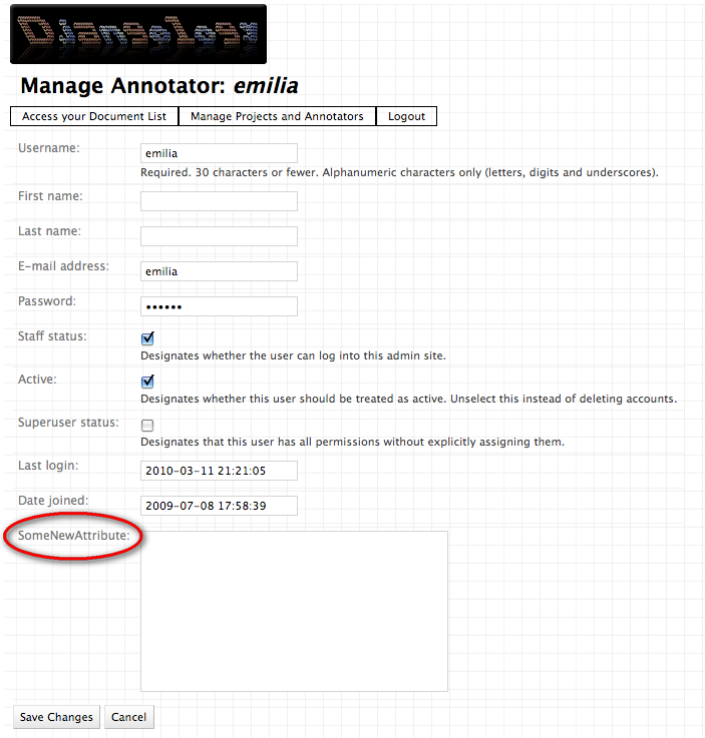
A Figures



(a) The Djangology database schema is transparently created and managed by the application.

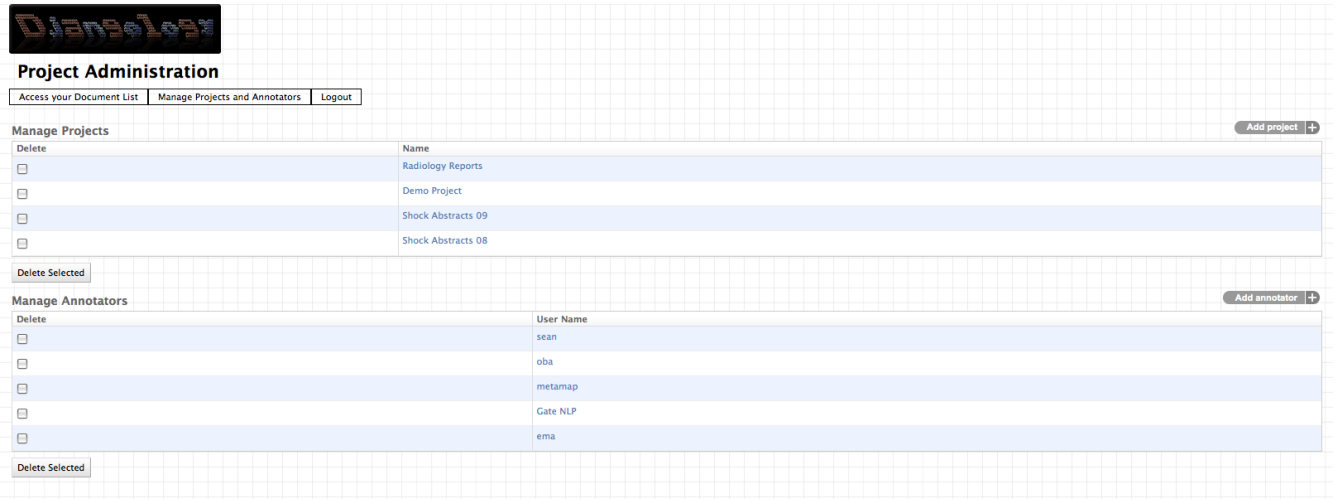
```
class Annotator(User):
    documents = models.ManyToManyField("Document", blank=True)
    admin = models.BooleanField()
    someNewAttribute = models.TextField()
    def __str__(self):
        return ''.join((self.first_name, self.last_name, '-' + self.username))
```

(b) Both the database schema and the web forms can be effortlessly extended by simply modifying the underlying Django model objects. In this example, a new field is added to annotator accounts.

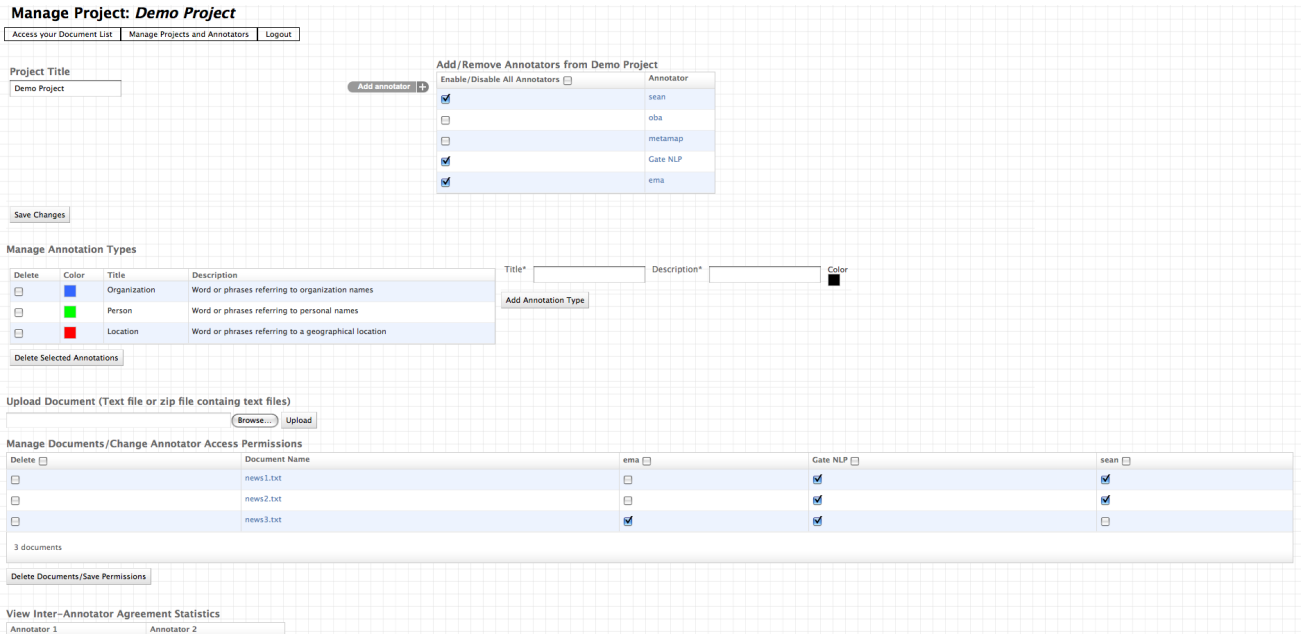


(c) Model object modification are immediately reflected in the web interface.

Figure 1: Djangology allows rapid customizations.

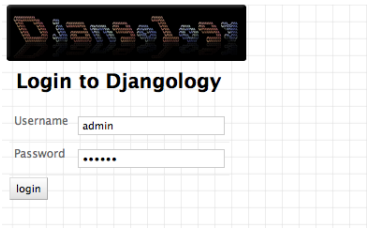


(a) A web interface allowing administrators to manage projects and users

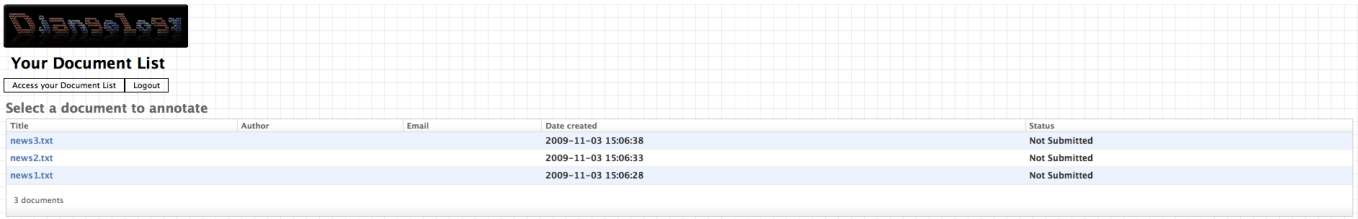


(b) A web interface allowing administrators to define/modify projects

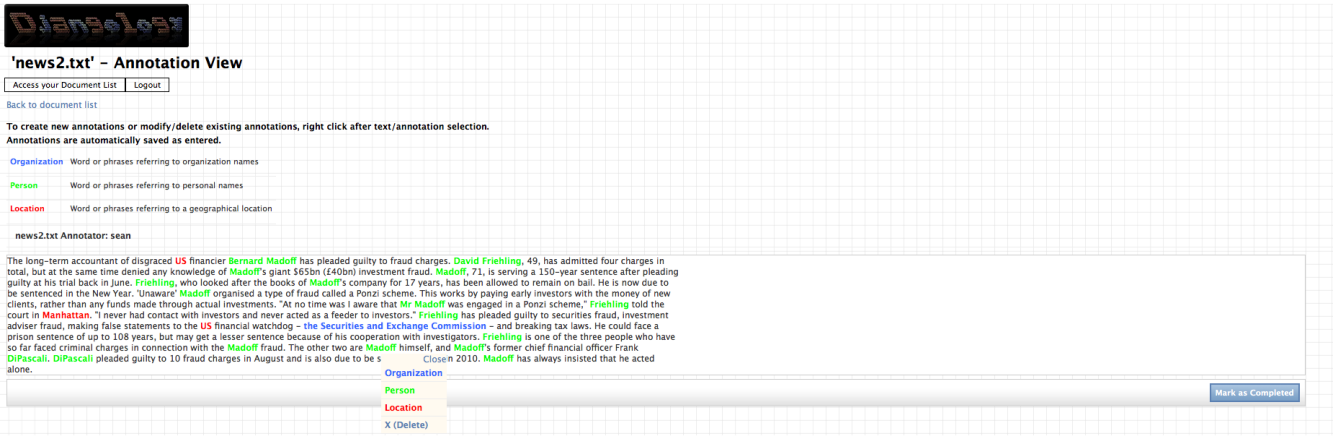
Figure 2: Sample pages demonstrating the Djangology administration interface



(a) Annotators are sent authentication information via email and a link to access the web application

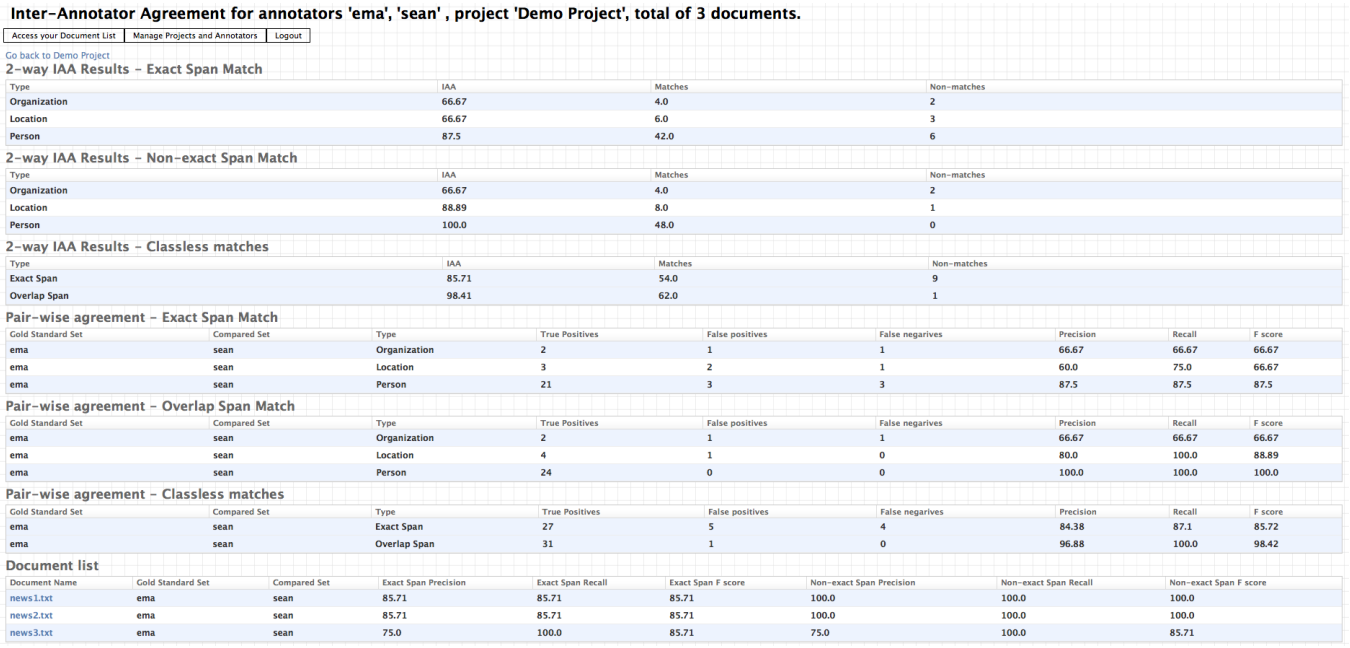


(b) List of documents assigned to a project contributor



(c) An intuitive interface allowing annotation editing based on the project annotation schema as defined by the project administrator

Figure 3: Sample pages demonstrating the interface used by project annotators



Inter-Annotator Agreement for annotators 'ema', 'sean' , project 'Demo Project', total of 3 documents.

Access your Document List | Manage Projects and Annotators | Logout

Go back to Demo Project

2-way IAA Results - Exact Span Match

Type	IAA	Matches	Non-matches
Organization	66.67	4.0	2
Location	66.67	6.0	3
Person	87.5	42.0	6

2-way IAA Results - Non-exact Span Match

Type	IAA	Matches	Non-matches
Organization	66.67	4.0	2
Location	88.89	8.0	1
Person	100.0	48.0	0

2-way IAA Results - Classless matches

Type	IAA	Matches	Non-matches
Exact Span	85.71	54.0	9
Overlap Span	98.41	62.0	1

Pair-wise agreement - Exact Span Match

Gold Standard Set	Compared Set	Type	True Positives	False positives	False negarives	Precision	Recall	F score
ema	sean	Organization	2	1	1	66.67	66.67	66.67
ema	sean	Location	3	2	1	60.0	75.0	66.67
ema	sean	Person	21	3	3	87.5	87.5	87.5

Pair-wise agreement - Overlap Span Match

Gold Standard Set	Compared Set	Type	True Positives	False positives	False negarives	Precision	Recall	F score
ema	sean	Organization	2	1	1	66.67	66.67	66.67
ema	sean	Location	4	1	0	80.0	100.0	88.89
ema	sean	Person	24	0	0	100.0	100.0	100.0

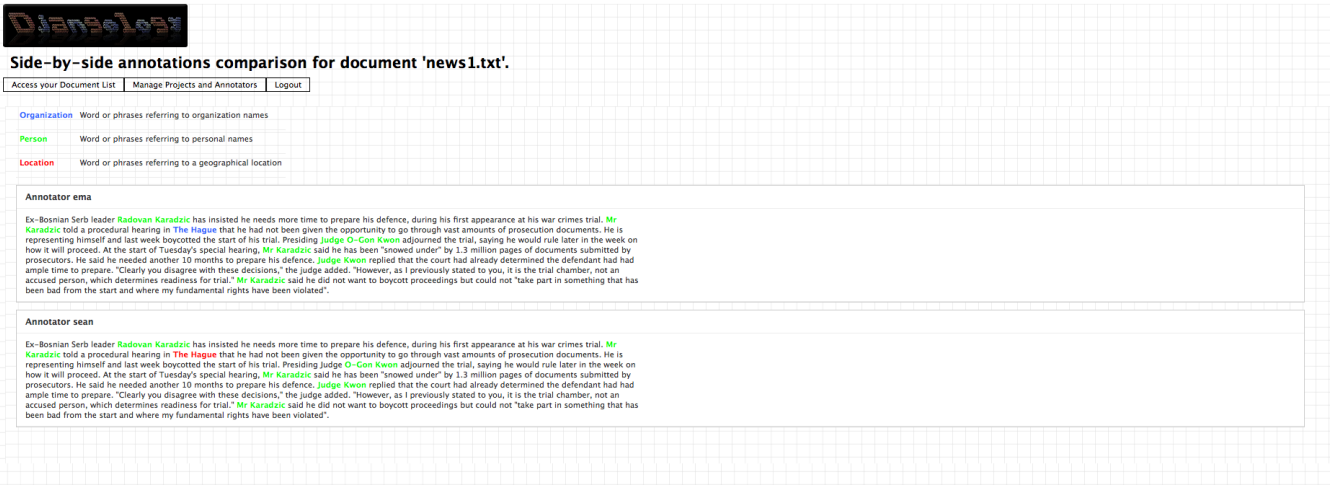
Pair-wise agreement - Classless matches

Gold Standard Set	Compared Set	Type	True Positives	False positives	False negarives	Precision	Recall	F score
ema	sean	Exact Span	27	5	4	84.38	87.1	85.72
ema	sean	Overlap Span	31	1	0	96.88	100.0	98.42

Document list

Document Name	Gold Standard Set	Compared Set	Exact Span Precision	Exact Span Recall	Exact Span F score	Non-exact Span Precision	Non-exact Span Recall	Non-exact Span F score
news1.txt	ema	sean	85.71	85.71	85.71	100.0	100.0	100.0
news2.txt	ema	sean	85.71	85.71	85.71	100.0	100.0	100.0
news3.txt	ema	sean	75.0	100.0	85.71	75.0	100.0	85.71

(a) Project administrators can view inter-annotator agreement statistics



Side-by-side annotations comparison for document 'news1.txt'.

Access your Document List | Manage Projects and Annotators | Logout

Organization Word or phrases referring to organization names

Person Word or phrases referring to personal names

Location Word or phrases referring to a geographical location

Annotator ema

Ex-Bosnian Serb leader Radovan Karadzic has insisted he needs more time to prepare his defence, during his first appearance at his war crimes trial. Mr Karadzic told a procedural hearing in The Hague that he had not been given the opportunity to go through vast amounts of prosecution documents. He is representing himself and last week boycotted the start of his trial. Presiding Judge O-Gon Kwon adjourned the trial, saying he would rule later in the week on how it will proceed. At the start of Tuesday's special hearing, Mr Karadzic said he has been "snowed under" by 1.3 million pages of documents submitted by prosecutors. He said he needed another 10 months to prepare his defence. Judge Kwon replied that the court had already determined the defendant had had ample time to prepare. "Clearly you disagree with these decisions," the judge added. "However, as I previously stated to you, it is the trial chamber, not an accused person, which determines readiness for trial." Mr Karadzic said he did not want to boycott proceedings but could not "take part in something that has been bad from the start and where my fundamental rights have been violated".

Annotator sean

Ex-Bosnian Serb leader Radovan Karadzic has insisted he needs more time to prepare his defence, during his first appearance at his war crimes trial. Mr Karadzic told a procedural hearing in The Hague that he had not been given the opportunity to go through vast amounts of prosecution documents. He is representing himself and last week boycotted the start of his trial. Presiding Judge O-Gon Kwon adjourned the trial, saying he would rule later in the week on how it will proceed. At the start of Tuesday's special hearing, Mr Karadzic said he has been "snowed under" by 1.3 million pages of documents submitted by prosecutors. He said he needed another 10 months to prepare his defence. Judge Kwon replied that the court had already determined the defendant had had ample time to prepare. "Clearly you disagree with these decisions," the judge added. "However, as I previously stated to you, it is the trial chamber, not an accused person, which determines readiness for trial." Mr Karadzic said he did not want to boycott proceedings but could not "take part in something that has been bad from the start and where my fundamental rights have been violated".

(b) Side-by-side document comparison page allowing project administrator to analyze sources of annotation disagreement

Figure 4: Inter-annotator agreement and error analysis tools