

Redcoat: A Collaborative Annotation Tool for Hierarchical Entity Typing

Michael Stewart^(✉), Wei Liu^(✉) and Rachel Cardell-Oliver

The University of Western Australia

35 Stirling Highway, Crawley, Western Australia

michael.stewart@research.uwa.edu.au, wei.liu@uwa.edu.au, and

rachel.cardell-oliver@uwa.edu.au

Abstract

We introduce Redcoat, a web-based annotation tool that supports collaborative hierarchical entity typing. As an annotation tool, Redcoat also facilitates knowledge elicitation by allowing the creation and continuous refinement of concept hierarchies during annotation. It aims to minimise not only annotation time but the time it takes for project creators to set up and distribute projects to annotators. Projects created using the web-based interface can be rapidly distributed to a list of email addresses. Redcoat handles the propagation of documents amongst annotators and automatically scales the annotation workload depending on the number of active annotators. In this paper we discuss these key features and outline Redcoat’s system architecture. We also highlight Redcoat’s unique benefits over existing annotation tools via a qualitative comparison.

1 Introduction

Recent successes of deep learning in natural language processing (NLP) is largely fuelled by high quality annotated datasets. Annotation tools provide the means to label data, and are vital for obtaining good results across a wide range of NLP tasks such as named entity recognition¹, question answering², and natural language inference³. One common underlying sub-component of these NLP tasks is entity typing, which involves identifying the type(s) of every entity in a document. Entity typing is also an enabling technique for utilising unstructured text in visualisation and knowledge discovery (Stewart et al., 2017).

Many recent entity typing taxonomies are structured in a hierarchy as opposed to a flat set of types. FIGER (Ling and Weld, 2012), for example, is a 112-class shallow hierarchy derived

from Freebase. TypeNet (Murty et al., 2017), derived from Freebase and Wordnet, features over 1900 types with an average depth of 7.8. It has been shown that incorporating these hierarchies into deep learning models improves classification accuracy (Murty et al., 2018; Ren et al., 2016).

Despite the needs of deep learning algorithms for labelled data with hierarchical entity types, a review of existing annotation tools shows there is *no support for multi-label tagging using hierarchical taxonomies*. Existing annotation tools, which are designed for the labelling of entity recognition data as opposed to entity typing data, only support one label per token. These systems also *do not allow for the modification of the taxonomy during annotation*. This lack of support is especially troublesome for real-world applications that are domain specific and typically with no standard category hierarchies. A tool that can leverage the annotation efforts as knowledge elicitation for domain taxonomy creation and refinement is very much in need.

While many annotation tools claim to maximise the speed of annotation, *few tools also optimise the time it takes for a project creator to set up and distribute an annotation project*. BRAT (Rapid Annotation Tool) (Stenetorp et al., 2012), the most popular annotation tool, requires project creators to read documentation, split their data into folders, set up a web server, and email their annotators links to their respective folders.

In light of these current issues, we introduce Redcoat, a web-based collaborative annotation tool for hierarchical entity typing. Redcoat was built with four primary goals in mind:

1. *Hierarchical*: Support entity hierarchies and multi-label annotation.
2. *Flexible*: Allow hierarchy refinement during annotation.

¹<https://www.i2b2.org/NLP/DataSets/>

²<https://rajpurkar.github.io/SQuAD-explorer/>

³<https://nlp.stanford.edu/projects/snli/>

3. *Rapid*: Reduce annotation time and time taken for project creation and distribution.
4. *Easy to use*: Keep it simple and intuitive for both annotator and project owner.

This paper is structured as follows. We begin by reviewing existing annotation tools. We then outline Redcoat’s key features, namely its intuitive and rapid project creation interface, annotation interface, and project dashboard. We describe Redcoat’s system architecture and then present a qualitative comparison between Redcoat and existing annotation tools. Finally, we provide a link to an online demonstration of Redcoat as well as a code repository link and demonstration video.

2 Related work

Among the several open source annotation tools available, the most popular is BRAT (Stenetorp et al., 2012), a web-based annotation tool that is designed to maximise annotation speed. BRAT supports the annotation of a wide variety of NLP tasks, including entity recognition, event extraction, and POS tagging. It also offers corpus search functionality.

GATE Teamware (Bontcheva et al., 2013) is another popular web-based annotation tool. It places a stronger emphasis on user management than BRAT, allowing for multiple user roles. It also provides automatic pre-processing of documents to improve annotation speed.

WebAnno (Yimam et al., 2013), based on the BRAT editor, features a strong emphasis on crowdsourcing via the CrowdFlower platform⁴. WebAnno also allows for the annotation of several NLP tasks. Unlike BRAT, however, WebAnno uses a relational database to model users, projects, documents, and tags. This provides useful features such as project monitoring and user management.

More recent annotation tools include SAWT (Samih et al., 2016), a lightweight web-based annotation tool that aims for simplicity and ease of use. Yedda (Yang et al., 2018) offers label recommendations via machine learning and provides both command line and web-based interfaces. SANTO (Hartung et al., 2018), which is designed primarily for slot-filling tasks, enables the formation of relational structures from an ontology. It also visualises the annotations of

⁴Crowdflower. <https://crowdfLOWER.com>

every user at once to help project owners monitor and curate the quality of the annotations. TALEN (Mayhew and Roth, 2018) is another recent tool that specialises in the annotation of low resource entities (i.e. where the annotators do not speak the language of the dataset). EasyTree (Tratz and Phan, 2018) is specifically designed for the annotation of dependency trees, and is integrated with Amazon Mechanical Turk crowdsourcing platform.

Several commercial annotation tools also exist, such as LightTag⁵, TagTog⁶, and Prodigy⁷. While these tools offer an array of features, their pricing can be prohibitive for researchers.

3 Redcoat - Key features

3.1 Intuitive and rapid project creation

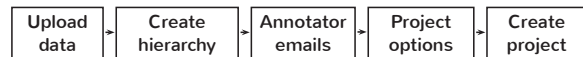


Figure 1: Redcoat’s project creation process.

One of Redcoat’s most notable features is its web-based project creation interface, which enables users to set up an annotation project and rapidly distribute it to a list of annotators. The process of project creation is shown in Figure 1. The project setup page allows for the user’s dataset to be dragged and dropped into a web-based form. The dataset is automatically tokenised by Redcoat prior to being stored in the database.

3.1.1 Hierarchical entity categories

Unlike many annotation tools, Redcoat supports the development of hierarchical entity categories and allows for each token to be labelled with more than one type. Users may specify their entity categories as either plain text with proper indentation, or as a hierarchy using an interactive tree diagram. Figure 2 shows an example hierarchy being built by the creator of an annotation project using the interface. Users may create, rename and delete categories by right clicking on categories within the tree diagram. Users may also simply paste their categories into a text box, denoting hierarchy levels via space characters, and the tree will automatically generate based on the given text.

⁵LightTag. <https://www.lighttag.io/>

⁶TagTog. <http://docs.tagtog.net/>

⁷Prodigy. <https://prodi.gy/>

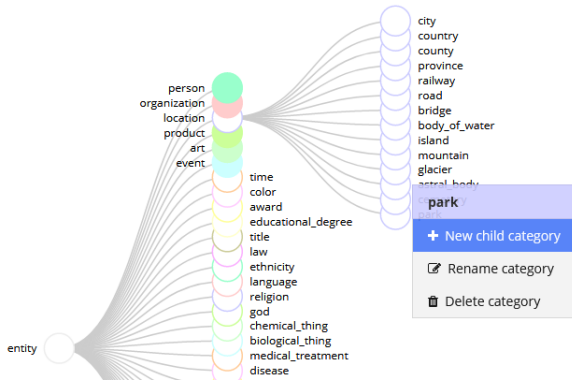


Figure 2: The category hierarchy generation window, which allows users to easily create, edit, and delete categories via an interactive tree diagram. In this example the user has loaded the FIGER preset and has right clicked on the “park” category to open the menu.

Redcoat also features three category hierarchy presets: *NER*, the standard Named Entity Recognition classes (PER, LOC, ORG, MISC), *FIGER* (Ling and Weld, 2012) (fine-grained entity recognition), and *Mining*, containing categories specific to workplace accident data in the mining industry. Selecting one of these presets via a dropdown menu instantly loads the corresponding hierarchy. UMLS⁸ and SNOMED CT⁹ are planned to be pre-loaded for medical dataset annotation.

3.1.2 Automatic project distribution

Project creators may specify a list of the email addresses of their annotators. Upon completion of the setup form, Redcoat sends an invitation to every valid email address in the list using Sendgrid¹⁰, a transactional email service. Users are invited to annotate the project regardless of whether they have registered for Redcoat, preventing the need for the project creator to coordinate the creation of user accounts.

3.1.3 Document propagation

In contrast to other annotation platforms, Redcoat automatically scales the annotation load of each annotator according to the number of users that have accepted their invitations to begin annotating. The documents are not split up into distinct sets, where each user has their own set of documents to annotate; they are instead distributed to annotators on a first-come-first-serve basis. The load of each annotator therefore depends entirely on how many

annotators are actively annotating the project. If, for example, a project creator specifies 10 email addresses on the setup form, but only 5 of them accept their invitations the next day, each annotator would be required to annotate 20% of the corpus. Once the remaining 5 users accept invitation, the load per annotator drops to 10%.

The project creator may also specify the “overlap”, i.e. the number of times each document should be annotated. An overlap of 2, for example, would mean each annotator would label 40% of the corpus (if 5 users have accepted invitation) and 20% of the corpus (if 10 users have accepted). Specifying an overlap value greater than 1 ensures more consistent data at the cost of annotation time.

3.2 Annotation interface

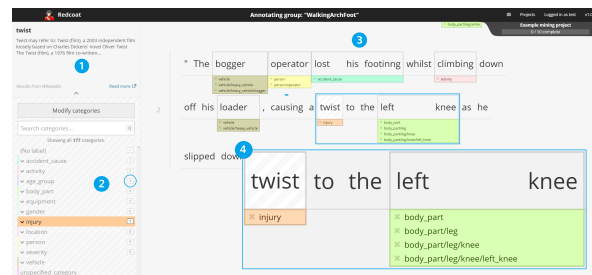


Figure 3: Redcoat’s simple annotation interface. (1) shows a Wikipedia summary of the selected token. (2) is the entity categories, which may be organised in a hierarchical structure and modified during annotation. (3) is the annotation interface. (4) shows a zoomed-in view of a selection of tokens.

Redcoat offers a simple annotation interface designed to maximise the speed of annotation. This interface is shown in Figure 3. The category hierarchy is displayed in the left menu, and categories may be expanded and minimised by clicking on them. Annotators may also search for categories using the built-in search menu.

The annotation interface allows for the use of both mouse and keyboard, providing annotators with a way to rapidly annotate documents if they elect to familiarise themselves with the hotkeys associated with navigating the documents (arrow keys) and the hierarchy (W, A, S, and D). The categories in the hierarchy also have associated numerical hotkeys, circled in Figure 3, aiming to speed up annotation.

Upon annotating a token, the token is automatically labelled with all of the label’s parent categories. Annotators may remove individual labels

⁸UMLS. <https://www.nlm.nih.gov/research/umls/>

⁹SNOMED CT. <http://www.snomed.org/>

¹⁰Sendgrid. <https://sendgrid.com/>

by clicking on the labels that appear underneath the annotated tokens.

The interface also presents an optional summary of the selected token taken from Wikipedia via the MediaWiki API¹¹ to reduce the need to Google search during annotation.

3.2.1 Modification of hierarchy

Redcoat allows for the modification of the category hierarchy during annotation. The extent to which the hierarchy should be modifiable is determined by the project creator. There are three options: *full permission*, whereby the hierarchy may be fully modified, *create only*, where annotators may only add new categories but may not delete them, and *no modification*. Deleted categories, along with their child categories, are removed from every annotation automatically.

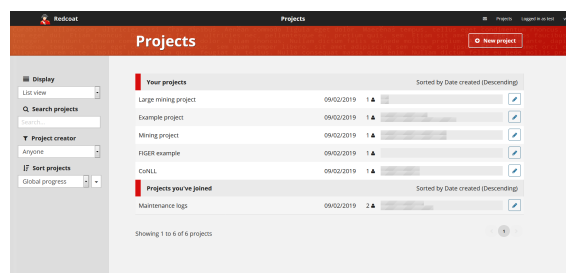
The ability to modify the hierarchy is useful for domain-specific datasets for which there are no standard category hierarchies. Project creators need not worry that their hierarchy does not contain every possible category in the dataset, as it may be updated dynamically. The flexible hierarchy allows for the development of categories to be an iterative process, thereby making the annotation process help with knowledge elicitation.

3.2.2 Automatic labelling

Redcoat provides an automatic labelling process to speed up annotation. Prior to presenting the documents to the annotator, any tokens that directly correspond to categories in the hierarchy are labelled with their corresponding type(s). For example, if a document contains the words “right arm”, and `body_part/arm/right_arm` is a category in the hierarchy, the token span will be labelled with `body_part/arm/right_arm`, `body_part/arm`, and `body_part`. Any incorrect labels may be deleted by the annotator. This process is implemented using regular expression parsing and does not noticeably affect load time.

3.3 Project dashboard

Redcoat’s project dashboard, as shown in Figure 4a, provides a way for project creators and annotators to quickly view all projects they’ve created or are currently annotating. The projects list may be sorted, filtered, and searched. Upon clicking a project users are presented with a detailed summary of the entire project. Project creators



(a) Redcoat’s Projects dashboard, which shows all projects the user has created and is involved in. Users may click on a project to bring up a detailed view of that project.

may view further details about their own projects, such as a list of pending/accepted invitations and a list of annotators that provides the ability to quickly download the completed annotations of the project.

3.3.1 Exporting annotations

Project creators may download annotated documents either per-annotator or for all annotators at once. At present these annotations are exported to the same JSON-based format used by state-of-the-art entity typing systems¹². The “download all” button compiles the annotations of every user into a dataset that contains the most commonly-assigned label for each token, providing project creators with a machine-learning-ready dataset with little effort.

4 System architecture

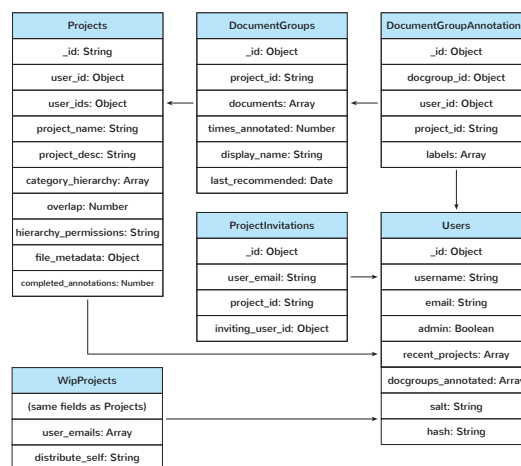


Figure 4: Redcoat’s underlying MongoDB schema.

Redcoat is written in Node.js¹³ and features an underlying MongoDB schema, shown in Figure 4.

¹¹MediaWiki API. <http://en.wikipedia.org/w/api.php>

¹²AFET. <https://github.com/INK-USC/AFET>

¹³Node.js. <https://nodejs.org>

System	Web-based project creation	Project monitoring	Curation feature	Document propagation	Class labels		
					Dynamic	Hierarchical	Multi-label
BRAT	✗	✗	✗	✗	✗	✓	✗
GATE	✓	✓	✓	✗	✗	✗	✗
SANTO	✓	✓	✗	✗	✗	✗	✗
SAWT	✓	✓	✗	✗	✗	✗	✗
YEDDA	✓	✓	✗	✗	✗	✗	✗
WebAnno	✓	✓	✓	✗	✗	✗	✗
Redcoat	✓	✓	✓	✓	✓	✓	✓

Table 1: A comparison of existing annotation tools with Redcoat. *Dynamic* refers to the ability for any user to modify the class labels during annotation.

The `Project` model stores information related to a project. `DocumentGroup` is a set of 10 documents belonging to a particular project. The documents are stored as arrays after tokenisation. `DocumentGroupAnnotation` stores the labels a particular user has assigned to a `DocumentGroup`. `ProjectInvitations` stores the invitations of a project, and is connected to the `User` table via `user_email` as opposed to `user_id` so that the invitation persists if the user has not yet registered. Finally, the `WIPProject` model stores information about a “work in progress” project, which is transferred to a new project upon completion of the setup form. This model allows for the data the user uploads to be persistent across refreshes and devices.

The category hierarchy is stored in the `Project` model as an array. Categories are stored in the form of strings, with different hierarchy levels represented by slashes (e.g. `person`, `person/boilermaker`). This array, along with every other field in each model, is subject to schema validation in order to ensure that the data is correctly stored. The category hierarchy, in particular, is validated both client and server side using a strict validation algorithm.

The majority of the front-end Javascript is written in jQuery¹⁴. The category hierarchy visualisation is implemented using D3.js¹⁵. Several other open-source libraries are used throughout the front-end, including DataTables¹⁶ and jsTree¹⁷.

5 Comparison with existing tools

Table 1 provides a qualitative comparison between Redcoat and other existing annotation tools.

¹⁴jQuery. <https://jquery.com>

¹⁵D3.js. <https://d3js.org>

¹⁶DataTables. <https://datatables.net>

¹⁷jsTree. <https://www.jstree.com>

Web-based project creation is present in all tools except BRAT, which requires data owners to split their dataset into multiple folders and place them into the appropriate location on their remote server. Consequently, *project monitoring* is also not present in BRAT, restricting the applicability of the system for real-world projects.

Few tools have a *curation feature*, allowing owners to specify the correct tags assigned to a token across tags provided by a set of annotators. Redcoat does not provide a formal curation interface, but includes automatic curation that selects commonly-assigned labels across annotators when downloading all annotations at once. This vastly simplifies the curation process and saves project creators considerable amounts of time.

Redcoat’s *document propagation* sets it apart from other tools. Annotation workload is automatically scaled depending on the number of active annotators, preventing the need to manually assign documents to annotators.

Redcoat also allows annotators to modify the class labels, and supports both hierarchical entity categories and multi-label annotation. Aside from BRAT’s ability to visualise label hierarchies, these features are not present in any other annotation tool.

Our qualitative analysis shows Redcoat is a highly flexible and powerful annotation tool, offering many benefits that distinguish it from other tools. It optimises speed, flexibility and ease of use while supporting hierarchical entity categories.

6 System demonstration

A demo of Redcoat is deployed online at <http://agent.csse.uwa.edu.au/redcoat/>. Users may create an account via the Register button on the homepage and set up a project immediately. A video of a demonstration of the system is available at <https://youtu.be/igtR8Sfi8oo>.

The source code is publicly available on GitHub¹⁸. The Readme file outlines how to set up Redcoat locally.

7 Conclusion and future work

In this paper we have introduced Redcoat, a collaborative annotation tool for hierarchical entity typing. It supports a variety of novel features, such as the ability to model entity categories as a hierarchy, label each token with more than one label, and update the hierarchy during annotation. Users may create projects using the web-based interface and quickly distribute their project to a list of email addresses. Redcoat handles the propagation of documents amongst users and automatically scales the annotation workload depending on the number of active annotators. These features distinguish Redcoat from existing annotation tools.

While Redcoat as presented here is ready to be used, there are still features under ongoing development. We are working on incorporating our deep-learning-based entity typing algorithms to make intelligent suggestions to support continuous automatic tagging. We also plan to visualise the annotation results and activity of annotators.

8 Acknowledgements

This research was funded by an Australian Postgraduate Award Scholarship and a UWA Safety Net Top-up Scholarship. The project is also supported by ARC DP150102405 and the NVIDIA GPU academic grant program.

References

- Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. *Gate teamware: a web-based, collaborative text annotation framework*. *Language Resources and Evaluation*, 47(4):1007–1029.
- Matthias Hartung, Hendrik ter Horst, Frank Grimm, Tim Diekmann, Roman Klinger, and Philipp Cimiano. 2018. *Santo: A web-based annotation tool for ontology-driven slot filling*. In *Proceedings of ACL 2018, System Demonstrations*, pages 68–73. Association for Computational Linguistics.
- Xiao Ling and Daniel S. Weld. 2012. *Fine-grained entity recognition*. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI’12, pages 94–100. AAAI Press.
- Stephen Mayhew and Dan Roth. 2018. *Talen: Tool for annotation of low-resource entities*. In *Proceedings of ACL 2018, System Demonstrations*, pages 80–86. Association for Computational Linguistics.
- Shikhar Murty, Patrick Verga, Luke Vilnis, and Andrew McCallum. 2017. *Finer grained entity typing with typenet*. *arXiv preprint arXiv:1711.05795*.
- Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. *Hierarchical losses and new resources for fine-grained entity typing and linking*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 97–109.
- Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. *Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding*. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378.
- Younes Samih, Wolfgang Maier, and Laura Kallmeyer. 2016. *Sawt: Sequence annotation web tool*. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 65–70.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. 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*, pages 102–107. Association for Computational Linguistics.
- Michael Stewart, Wei Liu, Rachel Cardell-Oliver, and Mark Griffin. 2017. *An interactive web-based toolset for knowledge discovery from short text log data*. In *International Conference on Advanced Data Mining and Applications*, pages 853–858. Springer.
- Stephen Tratz and Nhien Phan. 2018. *A web-based system for crowd-in-the-loop dependency treebanking*. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.
- Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2018. *Yedda: A lightweight collaborative text span annotation tool*. In *Proceedings of ACL 2018, System Demonstrations*, pages 31–36. Association for Computational Linguistics.
- Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 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*, pages 1–6.

¹⁸<https://github.com/Michael-Stewart-Webdev/redcoat>