EntiTies: An Interface for Annotating Ties between Entities in Text

Henry Feild
hfeild@endicott.edu
Computer Science Department
Endicott College
Beverly, MA, USA

Timothy Amello
tamel119@mail.endicott.edu
Computer Science Department
Endicott College
Beverly, MA, USA

Philip Lombardo
plombard@endicott.edu
Mathematics Department
Endicott College
Beverly, MA, USA

ABSTRACT

We describe an open source web application called EntiTies for annotating the entities throughout a text and the ties between them. Example uses of this tool include extracting character networks from novels or entity networks from legal documents. EntiTies' interface allows users to annotate a text from scratch, to process it automatically, or to process it automatically and then correct it. Forking allows multiple annotations to exist for the same text. EntiTies enables texts and annotations to be easily shared with others or kept private.

CCS CONCEPTS

• **Information systems → Data mining**; **Users and interactive retrieval**; • **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

entity extraction; entity network visualization; entity relationship extraction; text mining; semi-automated annotation

ACM Reference Format:
Henry Feild, Timothy Amello, and Philip Lombardo. 2020. EntiTies: An Interface for Annotating Ties between Entities in Text. In *2020 Conference on Human Information Interaction and Retrieval (CHIIR'20), March 14–18, 2020, Vancouver, BC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3343413.3377940

1 INTRODUCTION

Visualizing and analyzing the information stored in a text as a network is useful in many fields: who are the most connected individuals in a historical document, which characters bridge social groups in a work of fiction, or how legal entities relate in a contract. There are many ways to extract these networks, from the tedious and difficult to share manual method (e.g., making margin notes in a copy of "Dracula", then transcribing that information to a spreadsheet, and then importing that into a network visualization tool) to the error-prone fully automated method (e.g., running a digital text through software that extracts entities and ties, then plots and analyzes the network). Software exists that assists users in performing these annotations digitally (see Section 5 for a comprehensive list), but are largely targeted at natural language processing practitioners and linguists.

To make network extraction more accessible, we developed a simple to use, open source web application called EntiTies.[1] EntiTies allow users to upload a text and engage in manual, semi-automated, or fully automated network extraction workflows. Four types of annotation data are supported: entities (e.g., "Peter"), alias groups (e.g., "Peter", "Peter Pan", and "Pan"), mentions (spans of text that reference an entity), and ties (connections between two mentions or entities). Networks are visualized in real-time and can also be exported for use in applications such as NodeXL[2] or Gephi.[3] Texts and annotations can be shared or kept private and annotations can be forked.

Both the manual and semi-automated workflows bring a number of information retrieval opportunities in the form of system suggestions when grouping entity aliases, assigning a mention in the text to an entity, and annotating ties between entities. We describe the current options available in Sections 2 and 4 and discuss some interesting future directions in Section 6.

While there are many potential use cases of EntiTies, we will focus our examples on the original motivation for this work: extracting character networks from novels. To show the versatility of the annotation interface during the live demonstration of EntiTies, we will supply a wide range of texts (novels, plays, legal documents, historical accounts, etc.) that attendees can try out. Attendees can use EntiTies on our demonstration laptop, or access the web application on their own device.

We describe the three primary workflows of EntiTies in Section 2, the importance of sharing annotations in Section 3, the annotation interface in Section 4, and related work in Section 5. We provide a summary and discuss future directions in Section 6.

2 WORKFLOWS

EntiTies supports three primary workflows for annotating a text: manual, automated, and semi-automated. All workflows start with the user logging in and selecting an existing text to annotate or uploading a new one, which is then tokenized (see Figure 1a). Upon selecting a text, a list of all the annotations of the text that the user can view or modify are listed (see Figure 1b). These are displayed in a nested structure to show the forking history of each annotation.

CHIIR '20, March 14–18, 2020, Vancouver, BC, Canada

ACM ISBN 978-1-4503-6892-6/20/03.
https://doi.org/10.1145/3343413.3377940

[1] The source code is available at https://github.com/hafeild/entities under an MIT license; a live version is available at https://entities.greenbirdlabs.dev.
[2] https://www.smrfoundation.org/nodexl/
[3] https://gephi.org/

All texts have a read-only root annotation called "Blank slate", and it is from this annotation that all others are forked.

In the manual workflow, the user forks the "Blank slate" annotation and then selects locations in the text to mark them as mention of an entity or ties[4] between two entities (Figure 1c shows the interface, which is described in depth in Section 4). Entities can be merged into alias groups using the entity panel on the left. This mirrors annotating a book in the physical world, wherein an annotator goes through a book marking up pages with character mentions and the ties between them. EntiTies improves on this by assisting the user with suggestions. For example, when the user selects text and clicks "Add mention", they are presented with a list of characters ranked such that characters mentioned most recently are listed first.

Users can also choose to have their text annotated automatically. Presently, this follows a two-stage process in which stages can be run together or independently. In the first stage, entities, alias groups, and mentions are identified. In the second stage, ties are extracted. To run only stage 1, or both stages together, the user must click the "Run automatic annotation" button on the annotations page and select the appropriate option. A user can also run the second stage over an existing annotation in which entities and mentions have been resolved by selecting the tie extraction option when forking. At present, EntiTies implements stage 1 using a slight adaptation of BookNLP[5] [4], a Java package for identifying characters and their mentions in fiction. In the second stage, ties are extracted using a simple algorithm: if two entities are mentioned within a window of n tokens, a tie is added between them. The user can adjust n, with 30 set as the default. We plan to add additional options for both stages, such as incorporating the relationship extraction methods described by Agarwal [1].

The third workflow is to correct automated annotations, which we call semi-automated annotation. In EntiTies, this is done by (1) running an automatic annotation, (2) forking that annotation, and (3) making corrections through the interface. This, in theory, should yield the best of both worlds—less effort than manual annotation, but higher accuracy than automated annotation—and is what primarily motivated the creation of EntiTies.

3 SHARING ANNOTATIONS

In our experience, disseminating annotations of texts has been problematic. Scholars generally do not post their annotations, and even if they do, they are likely annotation artifacts (e.g., the resulting network) rather than in-line annotations (i.e., with exact location information within the text). Even with location information recorded, there are problems if the source texts differ. We encountered this when working with annotations a colleague made in a physical copy of a book. He made margin notes when characters in a novel spoke to each other for the first time. He then transcribed these to a digital spreadsheet, including the characters and the page on which they spoke. We were interested in extending this annotation to *anytime* two characters spoke (not just the first time) in a digital version of the book that lacked page numbers; while we did not ultimately carry this out, it might well have been easier to start over than to go through and align the original annotations with the digital version of the book.

Just as with any dataset, sharing annotations is important for several reasons. First, it allows others to inspect the underlying data. For example, a literary scholar made a claim using a window-based tie extractor that only considered ties that occurred at least three times [15]. That may seem like a reasonable way to eliminate noise, but at what cost? What ties are missing? To answer that requires implementing their algorithm and running it over the text—a considerable hurdle, especially for those without sufficient programming experience. Second, sharing annotations allows others to verify claims being made about the annotated text: is that character really the only bridge between two social groups in the text? Third and finally, sharing annotations and allowing them to be easily forked means others can extend or modify an annotation to support other analyses. For example, we may want to update our colleague's annotation to include ties whenever two characters speak to each other, not just the first time, in order to weight ties more heavily between characters with more speaking interactions.

To this end, EntiTies provides users with access management at both the text and annotation levels. Users can keep their texts and annotations private, grant or revoke specific users read or write access, or make them publicly viewable. Users can fork existing annotations they have view access to and make further changes.

4 ANNOTATION INTERFACE

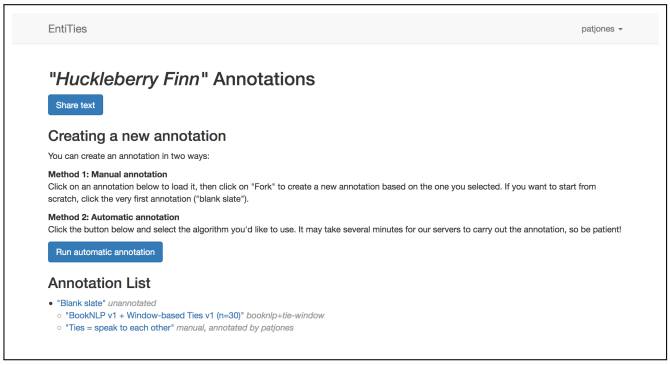
EntiTies provides a variety of interfaces that enable users to annotate four primary components in a text: entities, entity aliases, mentions, and ties. This annotation data is visualized in three distinct sections of the interface as shown in Figure 1c. In the left-most column of the interface, entities grouped with aliases are displayed in a checklist style format. The middle column is dedicated to the text itself; users can scroll through the entirety of their uploaded text in this section. Entity mentions and tie locations are highlighted, with more information available by clicking on the element. The right-most column of the interface houses the network, where each alias group is a node and each edge represents one or more ties between those groups; each edge is weighted as the sum of ties. The network is updated as the user engages with the interface (see below). Hovering over a node displays the name of the entity alias group. Nodes can be moved around and their locations frozen to make analyzing the network easier. An export button allows users to download the network in tab separated value or graphML [6] formats.

EntiTies provides several ways for users to create, modify, and delete annotation data. In addition to displaying the entities and their alias groups, the left-most column enables users to create new alias groups, disband existing ones, and move entities between alias groups. Each alias group is assigned a distinct color, and this color is used to represent that group across all columns.

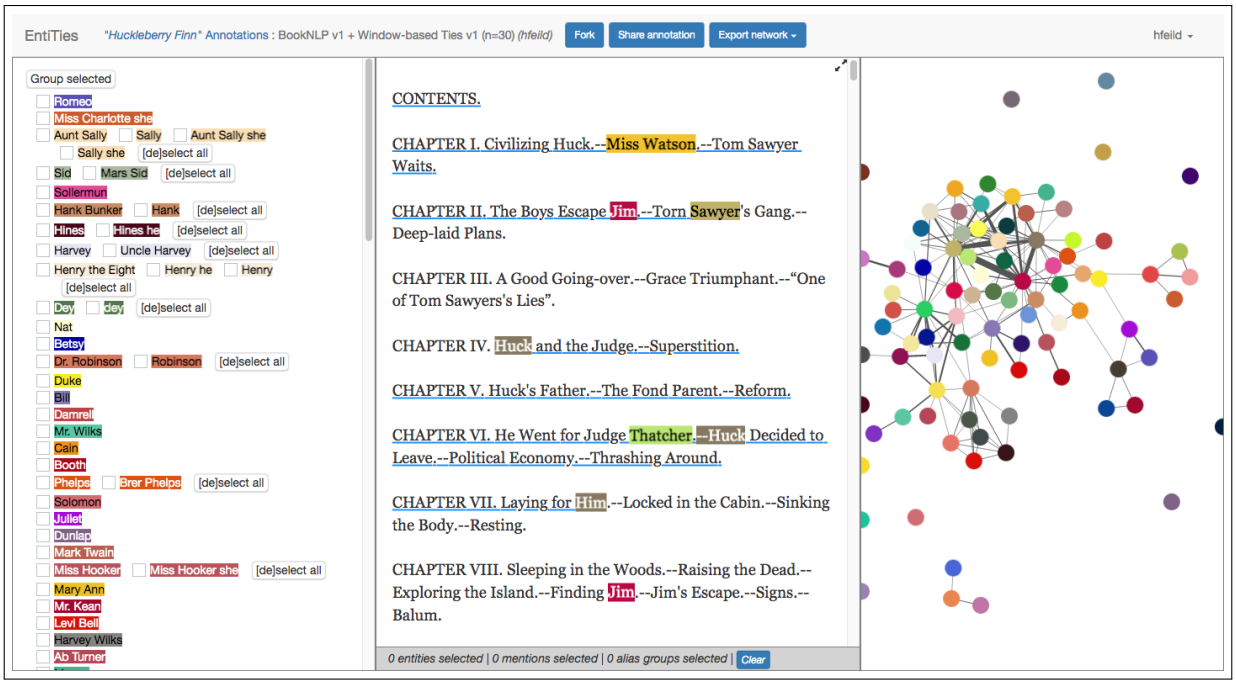
The middle column hosts the primary manipulation interfaces and all annotation data can be manipulated from this column. When users select a span of text, it causes a context menu of potential options to display: "Add Entity", "Add Mention", and "Add Tie".

[4] We use the social science term *tie* throughout this work to be consistent with the name Enti*Ties*; other common terms include *relationship*, *edge*, and *link*.
[5] See https://github.com/dbamman/book-nlp.
[6] http://graphml.graphdrawing.org/

(a) Homepage after logging in showing texts the user has uploaded as well as a list of all texts the user has permission to view.



(b) List of annotations for the currently selected text. Annotations are indented under the annotation they were forked from.



(c) The annotation interface, with entities organized into alias groups (left column), the text with mentions and ties highlighted (middle column), and the entity network (right column).

Figure 1: EntiTies screenshots.

Clicking "Add Entity" creates a new entity with a name corresponding to the selected text and is placed in its own alias group. A mention for that entity is also placed at the location of the selection. Clicking "Add Mention" displays a modal containing a list of entities in their alias groups with a section at the top showing the ten most recently mentioned entities and their aliases. Each has a radio button; the user must select one entity to associate with the selected mention location. Clicking "Add Tie" launches a modal that shows a paragraph of text surrounding the selection, with the selection highlighted. The user must select the two mentions or entities involved in the tie from a source and target entity dropdown menu, provide a label for the tie, and optionally supply a weight and indicate if the tie is directed. For convenience, the source and target entities are auto-populated with the first entity mention occurring prior and subsequent to the selection, respectively. The label is auto-populated with the selected text.

Clicking an existing mention displays a context menu with three submenus: "This Mention", "This Entity", "All aliases". The first option provides additional sub-options to delete the mention or reassign it to a different entity using the same modal as is presented when adding a new mention. The second option provides the sub-options to delete the entity associated with the mention or to move the associated entity to a new alias group. The latter launches a modal that displays each entity alias group, each with a radio button. The user can select which group to associate with the entity. The final option, "All aliases", provides two sub-options: delete the

alias group associated with the mention, or rename it. Selecting the rename option causes a modal to appear with a text box in which the user may enter the new name of the alias group. Clicking multiple mentions in sequence adds a fourth option to the context menu, "Selected", which has a submenu with options to delete the the alias groups associated with the selection, or combine all selected entities into a single alias group.

Clicking on an existing tie causes a context menu with two options to appear: delete the tie or edit it. Selecting the edit option causes the same modal used for tie creation to appear.

All of these interfaces work in tandem to allow for a complete annotation of the text, whether starting from an unannotated or previously annotated text.

5 RELATED WORK

Researchers in the areas of natural language processing and digital humanities have considered fully and semi-automated solutions for extracting entity networks from texts. Elson, Dames, and McKeown [8] describe a method for extracting speech based networks from novels that (1) identifies entities and mentions using named entity recognition software and clustering, (2) detects quoted speech and attributes it to an entity, and (3) extracts ties when quoted speech from two characters is found within 300 words of each other and with no other quoted speech occurs in between. Ties were identified with high precision but low recall (precision=0.95, recall=0.51, F1=0.67). Sack [15] considers narrative generation by examining triads in character networks. He extracted those networks by finding locations of character names and using a window-based tie extraction with $n = 10$, eliminating ties that occurred fewer than three times. Agarwal, Kotalwar, and Rambow report an F1 score of 0.68 for a system that extracts ties from gold standard entity mention annotations [2]. Other work on extracting entity ties found precision and recall under 50% [9, 16]. Agrawal [1] provides an overview of some of the related literature on entity network extraction techniques for novels, emails, and screenplays. In the few works that assess the accuracy of the algorithms put forward against ground truth human annotations, the effectiveness is relatively low (hence the need for EntiTie's semi-automated workflows). For a detailed survey of 127 works related to automatic social network extraction from works of fiction, see Labatut and Bost [11]. These methods can be implemented and incorporated into EntiTies as automatic annotation options; the EntiTies interface is agnostic to the methods used to identify entities, alias groups, mentions, and ties.

There are a number of text annotation tools related to EntiTies, such as brat[7], Callisto[8] [6], WebAnno[9] [7], CorefAnnotator[10] [14], ANNIS[11] [10], MMAX2[12] [12], UAM CorpusTool[13] [13] and many more. For instance, the brat rapid annotation tool and WebAnno are web applications for making and visualizing annotations of a text in a collaborative manner. Callisto, a Java-based linguistic annotation tool developed by MITRE between 2003 and 2013, can be used to mark and resolve named entities as well as passages in which relationships between entities are established. What sets EntiTies apart from these kinds of annotation systems is EntiTies' focus on the specific task of entity network extraction through locating entities, their mentions, and the ties between them. The entity and network panels in EntiTies are critical to this goal and are not present in these related systems.

Agarwal et al. [3] describe Sinnet, a now defunct demonstration web application with a Java backend that takes either raw text (pasted in) or text that is already annotated with named entities. In the first case, the software extracts named entities and performs named entity resolution over them. In both cases, the software extracts a social network. The user may choose a number of different tie extraction methods to apply (lexical, syntactic, semantic, or any combination thereof). The social network is then displayed and can be downloaded as either a graph modeling language (.gml) or .net [5] file. EntiTies aims to combine an annotation system like Callisto or brat and Sinnet in order to allow semi-automated annotations of texts to produce high-confidence social network extraction.

6 CONCLUSION

We described EntiTies, an open source web application for annotating texts with entities, their mentions in the text, and the ties between them. The uses of the extracted networks range from literary analysis of fictional works to visualizing the connections between entities in legal documents. EntiTies supports sharing texts and annotations to allow easy dissemination and to support further annotation by others. We also described three primary workflows supported by EntiTies: manual, automated, and semi-automated annotation.

Future work includes improving the current interface to make it easier and faster to annotate texts in a semi-automated fashion. Part of this includes presenting users with more relevant choices when clicking on an entity, entity mention, or tie location by leveraging information retrieval techniques.

EntiTies currently supports one algorithm for each of the two stages of automatic processing: BookNLP for entity and mention annotations and a basic window-based algorithm for tie extraction. Future work includes incorporating additional options for each stage, including a rich array of tie extraction algorithms that support identifying different kinds of ties (see the work of Agarwal [1]).

The semi-automated workflow described in Section 2 raises a few interesting research questions to be answered in future work. First, is semi-automated annotation more efficient than the manual workflow and more accurate than the fully automated workflow? A well-designed user study would address this question. Second, how can we propagate user corrections through the automatic annotation pipeline and update an annotation in real time? Addressing this will require designing new processing pipelines and very careful user experience design followed by extensive testing.

ACKNOWLEDGMENTS

We would like to thank Sam Alexander for inspiring this software and his detailed comments throughout its initial development.

[7] http://brat.nlplab.org
[8] http://mitre.github.io/callisto
[9] https://webanno.github.io
[10] https://github.com/nilsreiter/CorefAnnotator
[11] http://corpus-tools.org/annis/
[12] http://mmax2.net
[13] http://www.corpustool.com

REFERENCES

[1] Apoorv Agarwal. 2016. *Social Network Extraction from Text.* Ph.D. Dissertation. Columbia University.

[2] Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. 2013. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 1202–1208.

[3] Apoorv Agarwal, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. 2013. Sinnet: Social interaction network extractor from text. *The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations* (2013), 33–36.

[4] David Bamman, Ted Underwood, and Noah A Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 370–379.

[5] Vladimir Batagelj and Andrej Mrvar. 1998. Pajek-program for large network analysis. *Connections* 21, 2 (1998), 47–57.

[6] David Day, Chad McHenry, Robyn Kozierok, and Laurel Riek. 2004. Callisto: A Configurable Annotation Workbench. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LRECâĂŹ04)*.

[7] Richard Eckart de Castilho, Eva Mujdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*. 76–84.

[8] David K Elson, Nicholas Dames, and Kathleen R McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 138–147.

[9] Amina Kadry and Laura Dietz. 2017. Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1149–1152.

[10] Thomas Krause and Amir Zeldes. 2014. ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities* 31, 1 (2014), 118–139.

[11] Vincent Labatut and Xavier Bost. 2019. Extraction and Analysis of Fictional Character Networks: A Survey. *ACM Comput. Surv.* 52, 5, Article 89 (Sept. 2019), 40 pages. https://doi.org/10.1145/3344548

[12] Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. *Corpus technology and language pedagogy: New resources, new tools, new methods* 3 (2006), 197–214.

[13] Mick OâĂŹDonnell. 2008. Demonstration of the UAM CorpusTool for text and image annotation. In *Proceedings of the ACL-08: HLT Demo Session*. 13–16.

[14] Nils Reiter. 2018. CorefAnnotator - A New Annotation Tool for Entity References. In *Abstracts of EADH: Data in the Digital Humanities*. https://doi.org/10.18419/opus-10144

[15] Graham Alexander Sack. 2014. Character Networks for Narrative Generation: Structural Balance Theory and the Emergence of Proto-Narratives. *Complexity and the Human Experience: Modeling Complexity in the Humanities and Social Sciences* (2014), 81.

[16] Michael Schuhmacher, Benjamin Roth, Simone Paolo Ponzetto, and Laura Dietz. 2016. Finding relevant relations in relevant documents. In *European Conference on Information Retrieval*. Springer, 654–660.