Crowdsourcing the historical record: knowledge base construction for Chinese history at scale

Donald Sturgeon
Assistant Professor
Department of Computer Science
Durham University
djs@dsturgeon.net

Abstract

The tasks of annotating historical primary source materials and systematically recording knowledge about historical entities have close conceptual relationships. Annotations can be leveraged to extract knowledge about entities, and knowledge about entities can be leveraged to aid in the efficient annotation of texts. Many current systems for annotation and knowledge base construction specialize on performing one of these two tasks in isolation, either using a static knowledge base to create annotations, or a static set of annotated texts to extract knowledge. This paper describes a crowdsourced approach in which both tasks are carried out in parallel, with a knowledge base continually expanded through multi-user contributions to the annotation task immediately and automatically feeding back to provide automated assistance with ongoing and future annotations.

Keywords

Annotation; Linked Open Data; knowledge bases; calendars

Background and motivation

A variety of scholar-led projects – including in particular the China Biographical Database (CBDB) project, Academia Sinica's Grand Secretariat Archives Project, and Dharma Drum's Person, Place, and Time Authority Databases – have assembled substantial amounts of structured data about entities involved in the Chinese historical record. Other large-scale and more generally focused projects – such as Wikidata and Wikipedia – have also produced substantial volumes of relevant machine-readable data. Shared identifiers are frequently used to connect the same entity across different projects, enabling ease of comparison as well as offering the prospect of combining data from multiple sources for large-scale analysis. Alongside these knowledge bases, annotation tools – such as Recogito and Markus – have been created, which assist in connecting mentions of entities in a text with identifiers pointing to the relevant entity in one or more knowledge bases, producing an annotated text that can be used for further analysis with other tools.

Annotation of primary source materials can be used for a variety of purposes, and the same text can legitimately be annotated in vastly different ways corresponding to different use cases and research applications. Nevertheless, there are many examples of annotation types that can be

expected to be reusable for many different purposes – such as references to particular people, places, dates, etc. Accurately creating these annotations is a time-consuming task, for which computer assistance is beneficial, but unsupervised annotation alone often introduces an unacceptable rate of error. For common types of annotation in historical sources, it would be desirable to efficiently pool the resources of human annotators so that annotations can be created by individuals and groups working independently without requiring close coordination, and the resulting annotated texts reused by future projects without the need for repeated annotation.

At the same time, knowledge base construction for historical data frequently relies upon extracting this knowledge from primary source materials – in some cases, the very same materials which would benefit from annotation. Given texts containing appropriate annotations, some knowledge can be extracted automatically; likewise, given a comprehensive knowledge base, a greater degree of automated assistance can be given to the annotator in the annotation task. Ideally, therefore, the two tasks should interact to form a virtuous cycle in which annotations improve the knowledge base, and improvements to the knowledge base assist with the annotation task.

One approach to achieving these goals is to integrate mechanisms for the creation of annotations and knowledge claims into existing digital library systems. This paper introduces an infrastructure for creating, sharing, and maintaining annotations and knowledge claims within the framework of the *Chinese Text Project,* a large-scale crowdsourced digital library containing over 30,000 premodern Chinese written works (Sturgeon 2019).[1] By integrating named entity annotation and linking to existing authority databases and the wider Linked Open Data ecosystem into an existing repository, this approach facilitates annotation by a distributed user community and makes annotations available not only for internal use within the repository to facilitate advanced functionality, but also as annotated texts in consistent formats for further analysis. Computer-assisted knowledge claim extraction is used to leverage the annotations created through this process, contributing to the construction of a precisely sourced knowledge base of historical data that immediately feeds back into the annotation process.

Implementation

In this implementation, four core components are used to achieve these goals. The first is a chunked, serialized representation of textual objects. This representation – in this project, an XML compatible format – is used to encode core textual content, together with various types of annotation relating to the content. These annotations include entity disambiguation (e.g. identifying a string of text as an instance of a proper name with a particular authority identifier) as well as annotations for other purposes (such as maintaining alignment with photographic representations of the source text, or including illustrations or data about variant characters). The second component is a version-controlled storage medium for maintaining these serialized representations over time in a multi-user environment. This component also provides indexing

[1] https://ctext.org/

of the full-text contents of the materials, as well as the annotations present in them, maintaining the validity of these indexes in real time as changes are made to the textual objects and their annotations. The third component is a version-controlled graph database for storing entity data, enabling the aggregation of information about entities independently from their appearance at a specific location in a particular text. In addition to recording important historical facts such as names, dates of birth and death, etc., as well as evidential citations for these facts, a key role of this component is to offer a unified way of connecting multiple authority sources to a given entity.

The final component is a client-side user interface for performing semi-automated entity annotation and knowledge extraction (Figure 1). Building on ideas from earlier systems such as Recogito (Rainer et. al 2015)[2] and MARKUS (Ho and De Weerdt 2014),[3] this interface takes as input an XML document containing text and annotations, and provides the user with a range of options for adding, editing, and linking annotations, as well as extracting and storing knowledge claims about entities. These annotations currently consist primarily of person names, place names, bureaucratic office titles, written works, eras, dynasties, and dates specified according to historical Chinese conventions using combinations of era names, reign years, months, and lunar days. Content can either be user-supplied, or loaded directly from the digital library via API; in both cases, existing annotations present in a supported XML format are preserved and made editable through the user interface. Annotation consists of two stages: first, identifying strings of text which refer to entities; second, determining which entities these strings actually refer to. To aid in the first of these tasks, a comprehensive list of potential candidates is prepared by the system, using data extracted from entities already defined in the entity database, as well as candidates with external Linked Open Data identifiers imported from other sources such as Wikidata[4] – including, where available, identifiers for projects such as the China Biographical Database, [5] Buddhist Studies Authority Database Project, [6] and Academia Sinica's Grand Secretariat Archives Project,[7] as well as Wikipedia. The annotation interface uses this data to mark the location of probable entity strings; having done so, it then queries the server for further information about those strings which were found in the document. Manual annotations, made by user selection with the mouse, proceed in much the same way. Moving on to the second task, as automatic annotation is inherently fallible and requires human review to obtain fully accurate results, each annotation in the client program maintains an attribute indicating whether it is in a confirmed or unconfirmed state. Automatic annotations are first created in an unconfirmed state; annotations present in the initial document (either uploaded or imported from the digital library by API) are set to a confirmed state. User workflow consists of examining unconfirmed annotations, associating them with the appropriate referents and confirming them, or alternatively deleting them and/or creating entirely new entities as needed. As part of this workflow, the system uses prior user annotation decisions to offer bulk approval options,

[2] https://recogito.pelagios.org/
[3] https://dh.chinese-empires.eu/markus/beta/
[4] https://www.wikidata.org/wiki/Wikidata:Main_Page
[5] https://projects.iq.harvard.edu/cbdb/home
[6] https://authority.dila.edu.tw/
[7] http://archive.ihp.sinica.edu.tw/

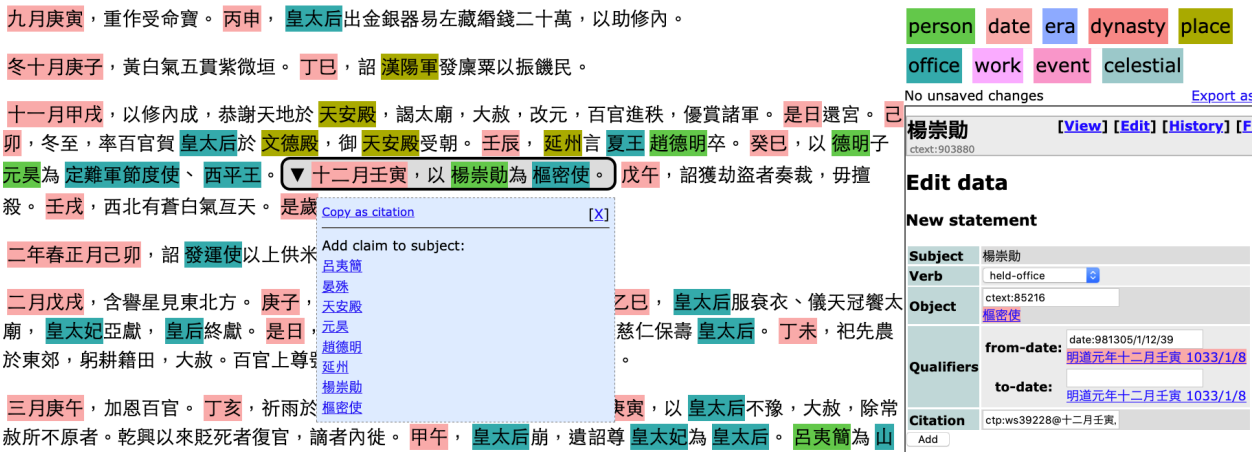
applying the same annotation (or implied analogous annotations) to subsequent candidates, avoiding the need to approve every annotation individually. Finally, the annotated document can be committed back to the version-controlled repository via API, and/or exported in various XML formats for use in other applications.

Figure 1: Entity annotation interface, showing linked data in the 15th chapter of the History of Song (宋史).

All entity references are treated identically in the annotation client; visual distinctions are made for the benefit of the user by displaying annotations color coded according to annotation type. The only exception is for date annotations, which in addition to specifying an entity (representing the era or ruler to which the date applies) can also specify offsets in terms of years, months, and days. This allows precise recording of date data without requiring prior or user-specified interpretation of complex calendar date information, while at the same time recording all dates in an unambiguous, machine-readable form, enabling their reliable interpretation and conversion to other calendar systems. For example, in the line from the *History of Song* reading "甲辰，以呂蒙正為太子太師、萊國公。", the date "甲辰" – though ambiguous when given here without its original context – clearly refers to day 甲辰 within the ninth month of the sixth year of the 咸平 era of 宋真宗. This date is therefore recorded as a multi-part annotation, explicitly recording each of these pieces of information: the era (specified as an entity reference, to avoid any possible ambiguity), the year (6), the month (9), and the day (41 in the 60-day cycle). As part of the annotation workflow, assistance is given to the user to enable efficient input: date references cascade through the text, with information from earlier dates (such as era, year, and/or month) providing suitable default values for subsequent dates. Assistance is provided for a variety of different date formats, including relative dates such as "the next day/year". The data produced through this process is sufficiently precise to facilitate exact and automatic mappings to the Gregorian and Julian calendars, by means of data published as part of the Time Authority Database (Bingenheimer et. al 2016) – in this example, resolving the date as being 6 February,

1005 AD in the Julian calendar. Machine-readable dates are used by the annotation client to request this information from a calendar server, and also to precisely record dates in knowledge claims in a format closely following their recorded form.[8]

Knowledge claim extraction can be carried out in parallel with the annotation task through the same interface. Each annotated entity mention points to an entity in the knowledge base; when a user annotates a new entity and indicates that it is not contained in the knowledge base, a new entity is created. At any point in the annotation workflow, the user can also add claims about entities to the knowledge base; these claims connect entities with other entities and/or various other types of recorded data, forming a knowledge graph primarily consisting of entities as nodes, and knowledge claims as edges. Claims follow a subject-verb-object model with the additional possibility of one or more qualifiers further expanding upon each claim – a similar data model to that used in Wikidata. Claims and qualifiers may also have evidence attached to them, in the form of a machine-readable reference to a specific string in a specified edition of a text. For example, the claim that Wang Anshi has a style name of "介夫" can be represented by a claim with the verb "name-style" and object the string "介夫"; this claim might have a citation from the 宋史 containing the text "王安石，字介甫，撫州臨川人。". The annotation client provides a workflow to aid users in creating this type of data efficiently. This starts with the user selecting any region of the text containing at least one annotation; this corresponds to the citation that will be attached to the knowledge claim. Next, the client suggests possible subject entities that the claim might be about – this will include any entities appearing within the citation, as well as other heuristic suggestions such as entities most frequently appearing in the document. The user then chooses a subject, and the system responds by listing verbs appropriate to the given entity type, together with appropriate object suggestions, and the machine-readable citation generated from the users chosen region of text (Figure 2). The user can choose any of the suggested options, or input their own, and the data can then be saved into the knowledge base.



[8] A large part of the motivation for doing this is the expectation that there will be errors in the primary sources, and also occasional errors and/or issues of conflicting or incomplete evidence involved in the complex calendar conversion process.

Figure 2: Automatic suggestions during manual knowledge claim input. An entity representing the office of 樞密使 has been suggested as the object of the verb "held-office"; a machine-readable date incorporating textual context corresponding to "明道一年十二月壬寅", and resolving to 1033/1/8 (Julian) has been suggested as the value for the "from-date" qualifier for this claim.

Also at any point in the process, the user can request that candidate knowledge claims are extracted automatically using the current state of the text and its annotations. Candidate knowledge inferences are made using regular expressions combined with annotation types, such that the textual content of an entity reference is not matched against the regular expression, but rather the general entity type is matched in its place. Knowledge claims can then be specified in terms of regex groups, where the group matching an entity reference is mapped not to the string, but to the entity identifier (Table 1). For example, a regex of the form "(<PERSON>)，<PLACE>?(<PLACE>)人也。" would match the (annotated) string "王守規，欒城人。"; this regex is itself associated with an inference specified in terms of its groups, e.g. "<group1> associated-place <group2>". In this case, the resulting knowledge claim would be that 王守規 is associated with the place 欒城.

Regular expression	**Knowledge claim inferred**
(<PERSON>)，<PLACE>?(<PLACE>)人也。	1 associated-place 2
以(<PERSON>)為(<OFFICE>)。	1 held-office 2
以(<DATE>)為(<EVENT>)。	2 date 1
(<DATE>)，(<PERSON>)薨。	2 died-date 1

Table 1: Examples of typed regular expressions used for knowledge extraction. Numbers in the second column represent regex groups. Regular expressions have been simplified for clarity of explanation.

This approach combines the flexibility of regular expressions with information from the annotation, allowing regular expressions to distinguish between what would otherwise be identical forms. For example, only the third regular expression of Table 1 would match the (appropriately annotated) string "以四月十四日為乾元節", inferring the claim that the event 乾元節 was held on 四月十四日; the structurally parallel string "以楊崇勛為樞密副使。" would instead only be matched by the second regular expression, leading to the inference that 楊崇勛 held the office of 樞密副使. Additional handling of date flow in the document can also be used, so that further qualifications can also be extracted – for instance, that the office was held by this person from some particular date. Because the generated claims are machine-readable, they can be automatically compared with data already contained in the knowledge base, and this information communicated to the user (Figure 3).

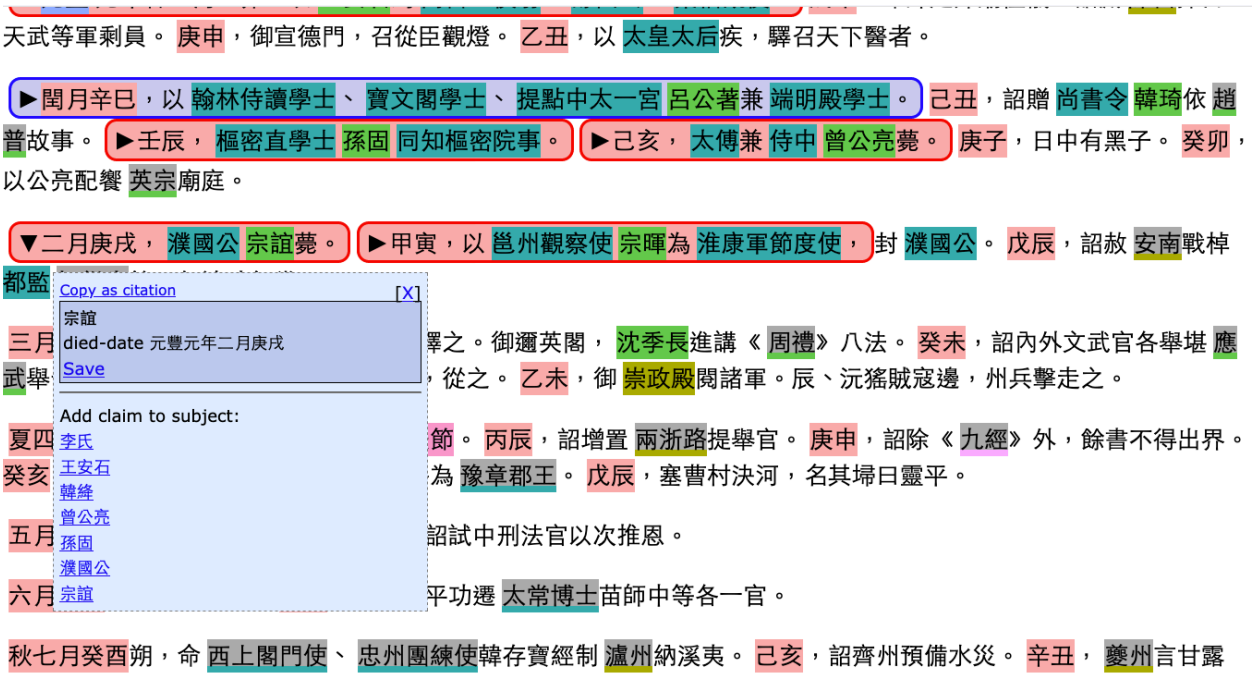


Figure 3: Automated knowledge extraction from a partially annotated text. Blue boxes indicate automatically extracted knowledge claims that are already contained in the knowledge base; red boxes indicate claims that have not yet been added. The opened claim invites the user to approve the addition of the extracted assertion: that 趙宗誼 died on 元豐元年二月庚戌 (March 21, 1078 AD).

Over time, this process of annotation and knowledge extraction leads to entity records with substantial amounts of machine-readable, precisely referenced historical data (Figure 4). As users expand the knowledge base with new entities, it also offers improved assistance with new texts. Both annotation and entity data can be accessed via API for use in other projects, and both can be used within the primary interface to the digital library to provide contextual assistance to readers of these texts (Figure 5).

王安石
ctext:855132

[View] [Edit] [History]

Relation	Target	Textual basis
type	person	
name	王安石	
name-style	介甫	《宋史·列傳第八十六》：王安石，字介甫，撫州臨川人。
associated-place	place:臨川縣	《宋史·列傳第八十六》：王安石，字介甫，撫州臨川人。
born-date	天禧辛酉年十一月十三日 1021/12/19	《能改齋漫錄·卷十議論》：王介甫辛酉十一月十三日辰時生，五十八歲，自首廳求出，知江寧府。
died-date	元祐元年四月癸巳 1086/5/21	《宋史·本紀第十七》：癸巳，王安石薨。
died-age	66	《宋史·列傳第八十六》：元祐元年，卒，年六十六，贈太傅。
father	person:王益	《宋史·列傳第八十六》：父益，都官員外郎。
authority-cbdb	1762	
authority-ddbc	A007519	
authority-viaf	49413576	
authority-wikidata	Q319618	
link-wikipedia_zh	王安石	
link-wikipedia_en	Wang_Anshi	
held-office	office:翰林學士	
from-date 治平四年九月戊戌 1067/11/2		《宋史·本紀第十四》：戊戌，以王安石為翰林學士。
held-office	office:參知政事	
from-date 熙寧二年二月庚子 1069/2/26		《宋史·本紀第十四》：庚子，以王安石參知政事。
held-office	office:同中書門下平章事	
from-date 熙寧三年十二月丁卯 1071/1/14		《宋史·本紀第十五》：丁卯，以韓絳、王安石並同中書門下平章事，

Screenshot

Figure 4: Part of the entity record for Wang Anshi 王安石, showing machine-readable knowledge claims and citations extracted using the annotation client.

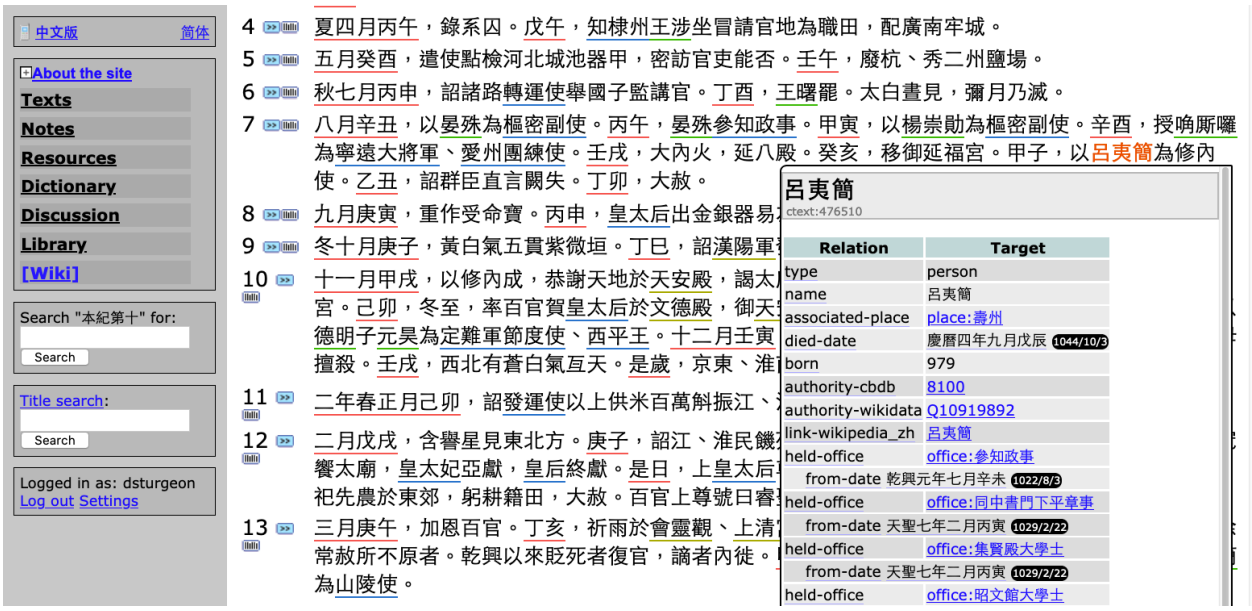


Figure 5: Contextual entity data displayed as part of the Chinese Text Project user interface.

The implementation is intentionally designed to be largely agnostic about the nature of entities and the behavior of different entity and annotation types – the only exception being date annotations. Information about the ontology of the knowledge base is recorded directly within it using the same data model: in particular, all properties (i.e. edge labels, such as “part-of” or “held-office”) are themselves entities of type “property”, and all qualifiers (such as “from-date”) are entities of type “qualifier”; the user interface itself has no built-in knowledge about specific

properties or qualifiers, and instead queries the knowledge base at runtime to dynamically discover what these are and how they should behave. As a result, it is expected that over time many additional entity types and knowledge claim types can be added, without requiring code modifications, as these additions themselves consist simply of modifications to the versioned graph database that can be made in exactly the same way as other content changes. The simplicity of the data model – and the absence of case-by-case processing – also mean that basic functionality such as edge-based search, and tabulation of references, can be easily implemented once and expected to work in a reasonable way with subsequently created entity types (Figures 6 and 7).

墨子
ctext:515227

Relation	Target	Textual basis
type	work	
name	墨子	
link-wikipedia_zh	墨子_(书)	
link-wikipedia_en	Mozi_(book)	
indexed-in	work:宋史	《宋史·志第一百五十八　藝文四》：《墨子》十五卷宋墨翟撰
juan-size 15		《宋史·志第一百五十八　藝文四》：《墨子》十五卷宋墨翟撰
stated-category 墨家		
creator	person:墨翟	《宋史·志第一百五十八　藝文四》：《墨子》十五卷宋墨翟撰
role-status 舊題		
indexed-in	work:四庫全書總目提要	《四庫全書總目提要》：《墨子》十五卷{{兩江總督採進本}}
juan-size 15		
stated-category 雜家		
edition 兩江總督採進本		
indexed-in	work:直齋書錄解題	《直齋書錄解題·直齋書錄解題·卷十》：《墨子》三卷
juan-size 3		
stated-category 墨家		
indexed-in	work:文淵閣書目	《文淵閣書目·文淵閣書目·卷二》：《墨子》一部一冊{{闕}}
bu-size 1		
ce-size 1		
indexed-in	work:百川書志	《百川書志》：墨子十五卷　宋大夫墨翟撰凡七十一篇
juan-size 15		
pian-size 71		
stated-category 墨家		
indexed-in	work:日本訪書志	《日本訪書志·日本訪書志卷七》：○《墨子》六卷{{萬歷辛巳書坊刊本}}
juan-size 6		
pian-size 53		
indexed-in	work:崇文總目	《崇文總目》：墨子十五卷 墨翟撰。

Figure 6: A fragment of the entity record for the work "墨子", showing links to information about this work and its editions as recorded in various historical collections.

宋史
ctext:545989

Relation	Target	Textual basis
type	work	
name	宋史	
ctext-work	ctp:work:wb975976	

Source	Relation	juan-size	stated-category
一司一務敕	indexed-in	30	刑法
三代地理志	indexed-in	6	地理
三十國春秋	indexed-in	30	霸史
三十國春秋鈔	indexed-in	1	霸史
三司條約	indexed-in	1	刑法
三國六朝攻守要論	indexed-in	10	史鈔
三國典略	indexed-in	20	編年
三國志	indexed-in	65	正史
三川古刻總目	indexed-in	1	目錄
三才定位圖	indexed-in	1	天文
三朝寶訓	indexed-in	30	別史
三楚新錄	indexed-in	3	霸史

Figure 7: A fragment of the entity record for the work "宋史", showing links to other works indexed within it. Precise textual evidence for each of these entries – as well as further data on the indexed work itself – is accessible through the corresponding entity record.

Conclusions and future work

The implementation described has been successfully deployed as part of the *Chinese Text Project* digital library, facilitating the collection, recording, and dissemination of entity annotations and knowledge claims. Much more work remains to be completed, including both increasing the coverage of annotations, and improving the efficiency with which annotations and additions to the knowledge base can be made.

The current annotation client implementation facilitates a practical crowdsourced workflow, and goes some way to incorporate intelligent sorting and filtering of entity candidate matches to reduce user workload. Firstly, temporal information is used to filter candidate entity matches, excluding entities known to significantly post-date a text being annotated – e.g. person names are automatically excluded from texts composed earlier to a person's likely date of birth. Secondly, existing confirmed annotations within a text are used to sort subsequent candidate lists, so that previously mentioned entities are treated as more probable matches for a subsequent entity for which they are a candidate, regardless of whether they are referred to by the same or a different name. Thirdly, edges from the knowledge graph are automatically used to sort adjacent candidates where these are associated by an edge in the knowledge graph, such that pairs of candidates with known associations are treated as more probable than others. This includes a variety of common cases, such as hierarchical place name information. For example,

in the string “開封祥符”, two matching candidate entity references “開封” and “祥符” are identified, each of which has multiple possible referents (such as 開封府 and 開封縣 for the former, and 祥符縣 and the era 大中祥符 for the latter). Since the knowledge graph contains an edge connecting 開封府 and 祥符縣 (in this case, an edge representing a “part-of” relationship), these two candidates will be preselected by the interface as the most probable pair, and the presence of this relationship visually indicated. This same simple rule applies similarly to many other commonly occurring edge types in the knowledge graph – such as title-person pairs (e.g. “申國公李穆” would automatically select the 李穆 from the Sui dynasty, as he held a title of 申國公, whereas “參知政事李穆” would instead select the Song dynasty 李穆 who served as 參知政事), and ruler-era pairs (as in the distinct but same-named 光天 eras of “漢光天” and “蜀光天”).

While these heuristics go some way to reducing effort in the annotation process, more sophisticated approaches to optimal candidate selection and ordering are possible. For example, broader information about the network structure of the knowledge graph itself can also be used to rank candidates – an approach which has been applied successfully in the closely related field of named entity disambiguation. Other available data – particularly date references – can also be leveraged to improve ranking accuracy. Over time as texts become increasingly comprehensively annotated, the scope for automated disambiguation of other similar materials will also increase, with the data produced through the manual annotation providing data suitable for training and evaluation of machine learning approaches.

References

Bingenheimer, Marcus; Hung, Jen-Jou; Wiles, Simon; and Zhang, Bo-yong. Modelling East Asian Calendars in an Open Source Authority Database. *International Journal of Humanities and Arts Computing* 10.2 (2016): 127–144.

Ho, Hou Ieong Brent, and De Weerdt, Hilde. MARKUS. Text Analysis and Reading Platform. 2014. http://dh.chinese-empires.eu/beta/

Simon, Rainer; Barker, Elton; Isaksen, Leif and de Soto Cañamares, Pau (2015). Linking early geospatial documents, one place at a time: annotation of geographic documents with Recogito. e-Perimetron, 10(2) pp. 49–59.

Sturgeon, Donald (2019). Chinese Text Project: A Dynamic Digital Library of Pre-modern Chinese. *Digital Scholarship in the Humanities* (Advance articles).

Sturgeon, Donald (forthcoming). Digitizing Premodern Text with the Chinese Text Project. *Journal of Chinese History*.

Vrandečić, Denny and Krötzsch, Markus. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, 57.10 (2014): 78–85.