Integrating Knowledge-Supported Search into the INCEpTION Annotation Platform

Beto Boullosa **Richard Eckart de Castilho** **Naveen Kumar**
Jan-Christoph Klie **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt, Germany
https://www.ukp.tu-darmstadt.de

Abstract

Annotating entity mentions and linking them to a knowledge resource are essential tasks in many domains. It disambiguates mentions, introduces cross-document coreferences, and the resources contribute extra information, e.g. taxonomic relations. Such tasks benefit from text annotation tools that integrate a search which covers the text, the annotations, as well as the knowledge resource. However, to the best of our knowledge, no current tools integrate knowledge-supported search as well as entity linking support. We address this gap by introducing knowledge-supported search functionality into the INCEpTION text annotation platform. In our approach, cross-document references are created by linking entity mentions to a knowledge base in the form of a structured hierarchical vocabulary. The resulting annotations are then indexed to enable fast and yet complex queries taking into account the text, the annotations, and the vocabulary structure.

1 Introduction

In many domains, annotating documents is a key requirement to solve complex problems like identifying sentiment targets in customer reviews, or identifying disease symptoms in medical texts. Traditionally, annotation tasks involved creating dense layers of annotation, e.g. part-of-speech or dependency annotations made on every single word, single or multi-token named entity mentions. Nowadays, the information to be annotated is often sparsely distributed, e.g. the mentions of particular types of entities. Finding spans of text which are candidates for a particular annotation type has thus become an important and challenging aspect of the annotation process. Therefore, it is essential that annotators can search the corpus, making queries over the full text as well as over the annotations. Linking entity mentions to a structured knowledge resource (e.g. a taxonomy) allows them to be disambiguated, which facilitates interpreting, processing, and navigating the annotated texts by effectively creating cross-document coreferences.

Consider a wine market specialist analysing a corpus of wine reviews. She wants to annotate mentions of different types of wines and link them to a knowledge resource, more specifically to a wine taxonomy. However, since annotating the entire corpus would take too much time, she wants to focus on statements made about certain properties of specific wines. Thus, she needs to search for keywords ("*price*", "*quality*", etc.), mentions of wines of certain types ("*Bordeaux*", "*Burgundy*"), or already annotated statements (e.g. to find comparative reviews). Thus, the specialist might pose queries such as "*sentences containing statements about the price of all kinds of Bordeaux wines*" in order to completely perform her corpus analysis. Note that the analyst cannot prepare a task-specific corpus in advance, because she only discovers which properties of the wines are addressed by the reviews as she goes along with the analysis.

We are not aware of any web-based text annotation tool that supports this kind of explorative annotation tasks requiring full-text search, cross-document entity linking, and annotation search, and, at the same time, takes into account the hierarchical relations of a taxonomy in a tightly integrated way. To address this gap, we integrate knowledge-supported search capabilities into the INCEpTION annotation platform (Klie et al., 2018) to provide a flexible way of searching the corpus during the annotation process. The corpus and annotations are indexed at token level. Primitive attributes (string, numeric, boolean) and attributes linking annotations to a knowledge base are indexed and can be queried. For linked annotations, it also considers the super-type/hypernym relations in the respective knowledge resource.

Section 2 highlights use cases in which those

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), pages 127–132
Brussels, Belgium, October 31–November 4, 2018.

functionalities are beneficial. Section 3 briefly introduces the INCEpTION annotation platform. Section 4 describes the knowledge-supported search functionality. Section 5 describes which types of knowledge resources the platform supports. Finally, Section 6 describes the related work.

2 Use cases

This section examines three exemplary scenarios of increasing complexity that highlight the benefits of knowledge-supported search in an annotation tool. We consider a wine market specialist who is investigating a corpus of wine reviews to identify the qualities most valued by the consumers and for which they may be willing to pay more. Her goal is to gain insights on consumer preferences, and the annotations she performs are a means to achieve this goal. The examples use the wine ontology from the W3C's *OWL Web Ontology Language Guide*,[1] a popular example of an OWL-based ontology.

Scenario I: Mention identification. The user wants to annotate mentions of a certain concept, e.g. types of wines. She starts with an initial list of wine types and uses the full text search to locate potential mentions, e.g. Bordeaux. Since the query is ambiguous (e.g. it could refer to the city or to the region instead of the wine type), she reviews each match and annotates it only when appropriate. If she discovers a wine type during this process that is not yet on her list, she adds it and again uses the full text search to locate and annotate its mentions.

Scenario II: Concept linking. The user now links the previously identified mentions to a taxonomy where the types of wines are organized into a tree or directed acyclic graph. For example, the vocabulary encodes that *Château d'Yquem* is a wine belonging to the *Sauternes* type, which in turn is a subtype of *Bordeaux*. These links effectively introduce cross-document coreferences within the corpus. Using the annotation search capabilities, the user wants to locate mentions of a wine type. This should consider the vocabulary structure, such that a search for a general wine type (e.g. Bordeaux) also finds mentions of all its subtypes.

Scenario III: Concepts in context. In addition to the linked concept mentions from the previous scenario, we assume that the corpus also carries other types of annotation, e.g. a custom claim annotation which identifies text spans containing statements made about properties of the wine. The user now wants to query the linked concept mentions in conjunction with these claims, e.g. to locate claims about particular types of wines. She may search for "*claims about wines either from the* Bordeaux *or from the* Burgundy *types, containing words matching the pattern* 'expensive.*'" (Figure 1).

These scenarios underline the benefit of integrating full text and knowledge-supported annotation search into an annotation tool. The next sections shows how INCEpTION addresses these needs.

3 The INCEpTION platform

INCEpTION[2] is a generic multi-user annotation platform aiming to cover three essential aspects of text annotation in a single tool: 1) corpus building, 2) knowledge modelling, and 3) annotation, and to combine them with machine-learning-based assistive mechanisms (so-called *recommenders*) to improve the annotation efficiency and quality.

INCEpTION is implemented as a Java-based web application using Tomcat, Spring Boot and Wicket. It is partially based on WebAnno (Eckart de Castilho et al., 2016), which we have modularized step-by-step to accommodate the needs of INCEpTION. This has allowed us to exclude certain WebAnno modules, e.g. the original automation module, which we replace with our own *recommender* framework, as well as to add new modules such as the search capabilities and knowledge base integration discussed here. We retain the WebAnno modules for project management, inter-annotator agreement calculation, adjudication, etc. as they are compatible with our new modules. The platform is open source software licensed under the Apache License 2.0.

This paper focusses on the annotation search capabilities of INCEpTION together with its knowledge base support. For the *recommender* mechanism, please refer to Klie et al. (2018).

4 Search

The search functionality of INCEpTION is accessible through a sidebar (1) in the annotation editor (Figure 1). It allows searching within the documents of the project the user is currently working on. After executing a query, the corresponding results are displayed grouped by document (2). Clicking on a result causes the annotation area to switch to the corresponding document/text span (3). Attributes that link an annotation to a knowledge

[1] https://www.w3.org/TR/owl-guide/wine.rdf

[2] https://inception-project.github.io

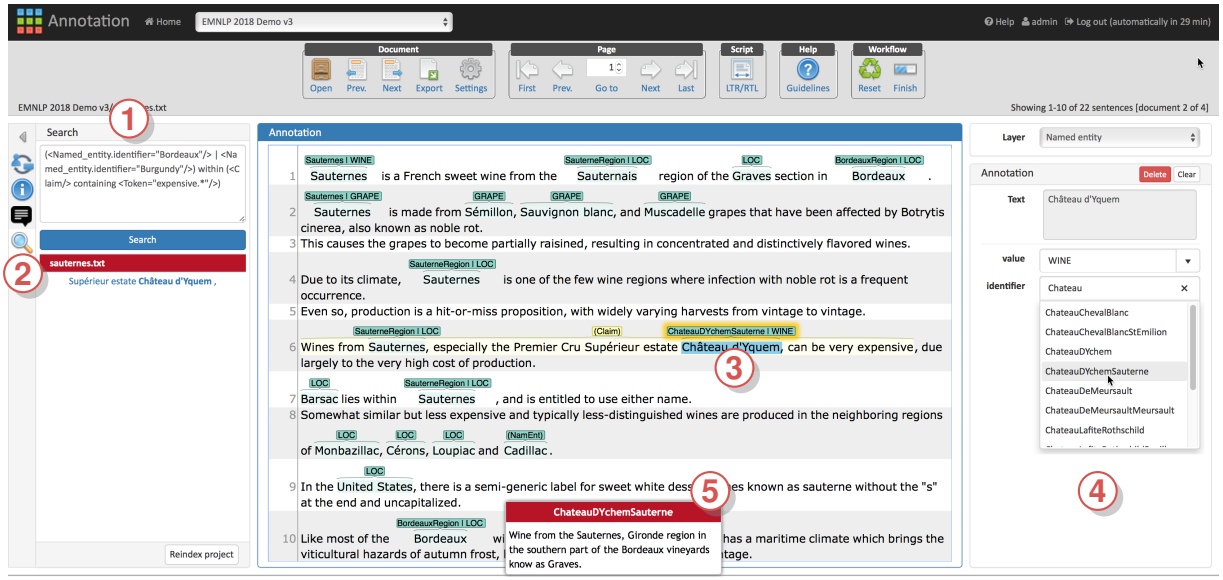

Figure 1: (1) Search sidebar with the query "*all mentions of wines belonging either to the Bordeaux or to the Burgundy type, located inside a claim which contains the pattern* expensive.* "; (2) search results grouped by document; (3) annotation area with a highlighted result; (4) auto-complete field allowing to select an entity from the knowledge base; (5) description of the entity the mouse cursor hovers over.

base item are conveniently editable through an auto-complete field (4).

4.1 Choosing a search framework

The knowledge-supported search functionality called for a search framework that met three requirements: 1) supporting text and annotation search; 2) supporting frequent updates, since the index needs to be updated whenever the user creates, changes or deletes an annotation; 3) it can be embedded directly in the annotation tool (i.e. no separate installation required). We considered three frameworks: the IMS Open Corpus Workbench, Mímir and MTAS.

The IMS Open Corpus Workbench (Christ, 1994) (IMS CWB) is an old but powerful tool to index and search annotated corpora. It introduced the popular Corpus Query Language (CQL).

Using Mímir (Tablan et al., 2015), queries over the annotated text can be combined with information from an knowledge base through SPARQL. This permits queries such as *find all mentions X of persons that were born in London*, where *X* is annotated as a person in the text, and *X was born in London* is contained in the knowledge base.

MTAS (Brouwer et al., 2017) is a recent framework which implements a large part of CQL on top of Apache Lucene.[3]

All frameworks support searching the full text as well as span annotations and their attributes.

Mímir and IMS CWB both assume that corpora are indexed once and queried often. Indexed documents can neither be updated nor easily be deleted and replaced. MTAS does not support updates to documents, but it allows deleting and then re-indexing individual documents.

IMS CWB is implemented in C and can be run either as a server or in an interactive mode. It cannot be easily embedded into a Java application such as INCEpTION. Mímir is implemented in Java, but its architectural design assumes that it is being used as a server. MTAS can be run as a server, but it can also be embedded into a Java-based application.

In conclusion, this made MTAS the best choice to be integrated with INCEpTION.

4.2 Integrating the search framework

To manage the annotations, INCEpTION uses UIMA (Ferrucci and Lally, 2004). For the knowledge base (KB), it uses RDF4J[4]. Thus, it was necessary to first implement a bridge from the UIMA data model to the MTAS data model while supporting the customizable layer configuration provided by INCEpTION. The ability to index annotation

[3] http://lucene.apache.org/

[4] http://rdf4j.org

attributes that link to KB items, i.e. classes and instances, was then added as a plugin to this bridge.

The bridge equally supports the built-in annotation layers (e.g. NAMED ENTITY) as well as user-defined layers (e.g. CLAIM). It indexes all the spans associated with all types of annotation layers (spans, relations, and chains). However, queries over relations and chains are limited since MTAS does not offer specific query operators for them. Indexed annotations must start and end at a token boundary. Subtoken annotations are not supported.

Each layer defines a set of attributes. E.g. the NAMED ENTITY layer defines a string attribute VALUE, which usually takes values such as LOC, PER, ORG and OTH for standard named entity annotation tasks. For our examples, we have also added WINE and GRAPE to that list. It also provides the attribute IDENTIFIER which can be used to link an annotation to a KB item (class or instance).

4.3 Full-text, annotation and attribute search

The token layer is built into INCEpTION and can be used to perform full-text queries. E.g., this query locates all occurrences of the token Bordeaux:

```
"Bordeaux"
```

Layers are referenced by their name. Attributes can be addressed using the syntax [layer].[attribute]. Assuming that wine mentions are annotated as named entities of type WINE, the following query finds all mentions of wines. This addresses the needs of Scenario I (Section 2).

```
<Named_entity.value="WINE"/>
```

4.4 Knowledge-supported search

Consider that the named entity annotation layer carries an IDENTIFIER attribute that holds the IRI (Internationalized Resource Identifier) of a KB item (Figure 2). These IRIs are included in the index, together with the IRIs of any items located higher in the ontology hierarchy. As IRIs are hard to read, the index also includes the human-readable labels associated with the entries, so that the user can query using these labels instead.

A KB item can either be a class in the ontology hierarchy (e.g. a wine type or subtype) or an instance (e.g. a specific wine). The following types of queries can be performed to search for annotations linked to the KB: 1) mentions of a specific KB item; 2) mentions of a KB item, including the mentions of its descendants.

The syntax for addressing the attributes linked to the knowledge base is the same as for normal attributes. The user can either match against the IRI of the linked KB item or against its label. This will retrieve all mentions of the given item, plus all mentions of its descendants in the ontology. Thus, the query effectively traverses the ontology hierarchy, starting in the given item and going down its corresponding subtree. This addresses queries like the one highlighted in Scenario II (Section 2).

```
<[layer].[attribute]="[label | IRI]" />
```

The following example matches all mentions of wines under the *Bordeaux* branch of the ontology:

```
<Named_entity.identifier="Bordeaux"/>
```

By appending -exact to the attribute name, it is possible to limit the query to mentions of exactly one particular item:

```
<[layer].[attribute]-exact="[label | IRI]" />
```

Note that multiple KB items may in principle carry the same label. To avoid this ambiguity, it may be necessary to query using the IRI.

Considering again that annotations are linked to the wine ontology, the following query locates all exact mentions of the *Clos de Vougeot* wine:

```
<Named_entity.identifier-exact =
  "http://www.w3.org/TR/2003/PR-owl-guide-
  20031209/wine#ClosDeVougeotCotesDOr"/>
```

The rich query language provided by MTAS allows to combine different query types like the ones previously introduced, using operators such as within or containing. Considering that our example dataset includes the custom CLAIM annotation type, we can address Scenario III (Section 2) by writing the following query, which retrieves all mentions of wines belonging to the Burgundy or Bordeaux types (and their subtypes), located inside a claim that matches the regular expression pattern expensive.* (Figure 1).

```
(<Named_entity.identifier="Burgundy"/> |
<Named_entity.identifier="Bordeaux"/>)
within (<Claim/> containing "expensive.*")
```

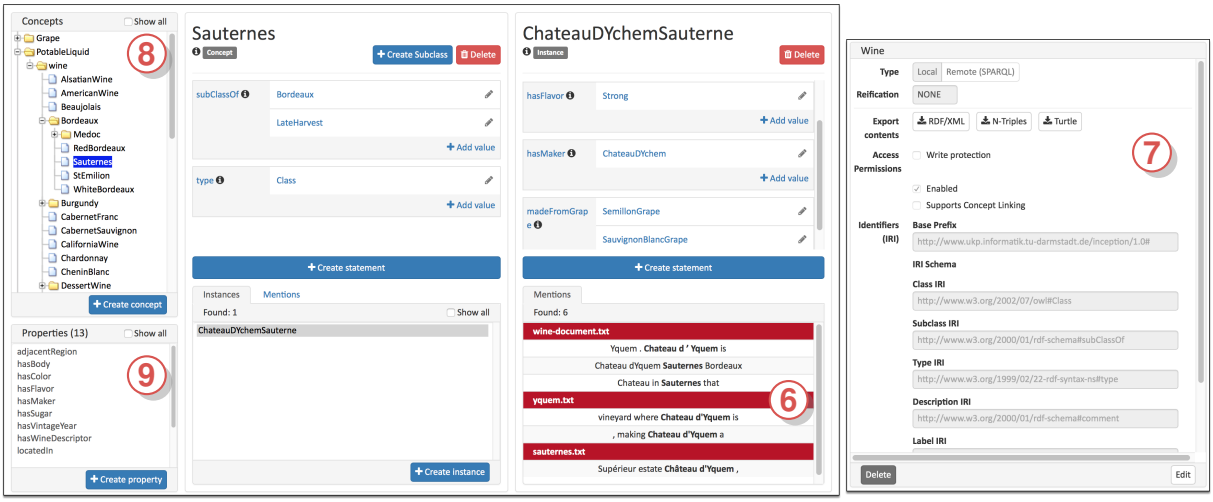

Figure 2: Knowledge base page (left): (8) concept explorer; (9) property explorer; (6) annotated mentions of the seleted KB item. Right: (7) mapping configuration editor.

5 Knowledge-base integration

The knowledge-oriented search capabilities of INCEpTION are enabled by its KB module. This module allows the user to create a KB from scratch or to import one from an RDF file. Remote KBs can be accessed in a read-only mode via the SPARQL.

The KB management page (Figure 2) allows editing classes, properties, instances and the corresponding statements they are defined by. Using the search module, it also displays any annotated mentions (6) of the currently selected KB item.

As the KB module is RDF-based, every piece of information is stored as a triple <subj, pred, obj>. Since this model is very abstract, there are a number of different schemas defining common identifiers (IRIs) that provide additional semantics, e.g. RDF Schema[5] uses the IRI rdfs:subClassOf to encode a subclass relation between the items identified by the subject and the object of a triple.

To support a broad range of different knowledge resources, INCEpTION offers a configurable mapping (7) (Figure 2). The user can choose from several predefined mappings (e.g. RDF, OWL, or SKOS) or define a custom mapping. The mapping mechanism relies on a minimal set of IRIs that must be defined for any KB used with the platform: the INSTANCE-OF relation is required to be able to identify instances, classes and properties within the ontology (<X, instance-of, Y>). Commonly rdf:type is used here, but e.g. the RDF version of Wikidata uses a different IRI. Additionally IRIs identifying CLASS and PROPERTY definitions are required in order to populate the concept explorer (8) and the property explorer (9) (Figure 2) - e.g. <X, instance-of, class>. The class hierarchy is defined via the SUBCLASS-OF IRI. Thus, hierarchies defined e.g. via rdfs:subClassOf or skos:broader are supported, but not hierarchies defined via skos:narrower.[6] While INCEpTION tries to detect root classes automatically, the corresponding query is resource intensive and may eventually time out on some large knowledge resources. Thus, it is also possible to bypass the automatic detection by manually specifying the IRIs of root classes. Finally, IRIs for LABELS and DESCRIPTIONS can be defined. If present, labels are used instead of the IRI when referring to a class, property or instance. Descriptions are shown as a tooltip (Figure 1) when linking an annotation to a KB item.

6 Related work

Several annotation tools support structured vocabularies or KBs and some can be used for cross-document annotation tasks. As INCEpTION is a generic annotation tool, we compare our work to the other generic tools.

WebAnno (Eckart de Castilho et al., 2016), while not offering explicit support for structured vocabularies, can approximate them by combining two of its features: tagsets and constraints. Constraints allow to show a certain attribute of an annotation only when another attribute has a specific value, e.g. to show a COUNTRY attribute only if the TYPE

[5] https://www.w3.org/TR/rdf-schema/

[6] https://www.w3.org/2004/02/skos/

property of the entity has the value location. Tagsets can then be used to control which values are acceptable for the entity type or country properties. However, WebAnno has no support for search.

AlvisAE (Papazian et al., 2012) supports linguistic and semantic annotations and can connect them to a structured vocabulary. However, it does not offer the ability to search over annotations and consequently also has no ability to make use of the vocabulary structure in such queries.

CROMER (Girardi et al., 2014) is a tool for entity and event coreference annotation. It allows to annotate and link entity mentions to entities defined in a knowledge base and in this way to create implicit cross-document coreference links. It also offers a simple string-based search to locate potential entity mentions. However, it does not allow to perform further searches involving the created annotations or the structure of the vocabulary.

NeuroCurator (O'Reilly et al., 2017) is a collaborative framework for annotating experiment parameters in scientific papers using an ontology-driven approach. It is rather an interactive knowledge base population tool than a tool for cross-document coreference. Queries over the texts that make use of the information of the KB are not possible.

7 Conclusion and Future Work

We have introduced a knowledge-supported search mechanism into a generic text annotation tool, INCEpTION, to support entity linking and cross-document coreference annotation tasks. The need for such a functionality was motivated using three scenarios, all of which are facilitated using the knowledge-supported search mechanism. In future work, we plan to further extend the search mechanism, e.g. allowing to search over annotation suggestions provided by the *recommender* framework of INCEpTION and by further enhancing the ability to match against information contained in the knowledge bases.

Acknowledgments

We thank Wei Ding, Peter Jiang, Marcel de Boer and Michael Bugert for their valuable contributions. This work was supported by the German Research Foundation under grant No. EC 503/1-1 and GU 798/21-1 (INCEpTION).

References

Matthijs Brouwer, Hennie Brugman, and Marc Kemps-Snijders. 2017. MTAS: A Solr/Lucene based Multi Tier Annotation Search solution. In *Selected papers from the CLARIN Annual Conference 2016, Aix-en-Provence, 26–28 October 2016*, 136, pages 19–37. Linköping University Electronic Press, Linköpings Universitet.

Richard Eckart de Castilho, Éva Mújdricza-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) at COLING 2016*, pages 76–84, Osaka, Japan.

Oli Christ. 1994. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of COMPLEX'94 3rd Conference on Computational Lexicography and Text Research*, pages 23–32, Budapest, Hungary.

David Ferrucci and Adam Lally. 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4):327–348.

Christian Girardi, Manuela Speranza, Rachele Sprugnoli, and Sara Tonelli. 2014. CROMER: a Tool for Cross-Document Event and Entity Coreference . In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 3204–3208, Reykjavik, Iceland. ELRA.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics - COLING 2018*, pages 5–9, Santa Fe, New-Mexico, USA.

Christian O'Reilly, Elisabetta Iavarone, and Sean L. Hill. 2017. A Framework for Collaborative Curation of Neuroscientific Literature. *Frontiers in Neuroinformatics*, 11(27):1–16.

Frèdèric Papazian, Robert Bossy, and Claire Nèdellec. 2012. AlvisAE: a collaborative Web text annotation editor for knowledge acquisition. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 149–152, Jeju, Republic of Korea. Association for Computational Linguistics.

Valentin Tablan, Kalina Bontcheva, Ian Roberts, and Hamish Cunningham. 2015. Mímir: An open-source semantic search framework for interactive information seeking and discovery. *Web Semantics: Science, Services and Agents on the World Wide Web*, 30(0):52–68.