University of Amsterdam

Requirements for enrichment tools

Boer, A.; Winkels, R.; Trompper, M.

Citation for published version (APA):
Boer, A., Winkels, R., & Trompper, M. (2016). *Requirements for enrichment tools*. (.Openlaws report; No. D2.2d1). Universiteit van Amsterdam. https://doi.org/10.5281/zenodo.158986

openlaws

Deliverable 2.2.d1

Requirements for enrichments tools

openlaws

JUST/2013/ACTION GRANTS

Grant Agreement Number 4562

Project Start date: 01.04.2014
Project End date: 31.03.2016

Report No. D2.2d1 – Requirements for enrichment tools

Version –   1.0

Document prepared by: UNIVERSITEIT VAN AMSTERDAM
                Alexander Boer, Radboud Winkels, Maarten Trompper

Contributors: -

Deliverable due date: 30.06.2014
Deliverable actual date: 30.06.2014

Document History

Date	Revision	Comments
30 June 2014	V1.0	Final version
24 June 2014	V0.3	Final draft for comments
10 June 2014	V0.2	
5 June 2014	V0.15	
May 2014	V0.1	First draft

Document Authors

Dr. Alexander Boer, dr. Radboud Winkels, Maarten Trompper BSc

Participant List

	Short Name	Organisation Name	Country
1	UVA	UNIVERSITEIT VAN AMSTERDAM	NL
2	SUSS	UNIVERSITY OF SUSSEX	GB
3	LSE	LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE	GB
4	ALP	ALPENITE SRL	IT
5	SUAS	FACHHOCHSCHULE SALZBURG GMBH	AT
6	BYW	BY WASS GMBH	AT

Executive Summary

This report gives a high level overview of requirements for Enrichment tools in the Openlaws.eu project. Openlaws.eu aims to initiate a platform and develop a vision for Big Open Legal Data (BOLD): an open framework for legislation, case law, and legal literature from across Europe.

Table of Contents

1 Introduction

Openlaws.eu aims to initiate a platform and develop a vision for Big Open Legal Data (BOLD): an open framework for legislation, case law, and legal literature from across Europe. Based on open data, open source software and open innovation principles we are adding a *social layer* to existing *legal information* systems. This document, the first deliverable of the project, lists requirements for the first version of the BOLD enrichment tools.

This document starts, in sections 2 and 3, with an abstract introduction into the objects that make up a Big Open Legal Data (BOLD) framework, in the present document called the *BOLD objects*, addressing both important aspects of big open legal data, and of the envisioned social networks that that will keep the process of enrichment going. Having introduced these, several sections follow that list detailed requirements for several of the BOLD objects identified.

1.1   Preliminaries on requirements for enrichment tools

Requirements for the enrichment tools are associated with BOLD objects, and are categorized into those that concern:

1. *Views* on BOLD objects
2. *Manipulations* of BOLD objects, and
3. *Process and provenance aspects* related to BOLD objects.

Objectives of the OpenLaws.eu project are 1) to develop an enrichment platform for BOLD objects, and 2) to outline a long term development path for enrichment of BOLD objects beyond the life and the means of the project. Requirements for the enrichment tools are therefore ambitiously formulated, with a limited ambition level within the project specified by a separate statement.

1.2   Preliminaries on BOLD objects

In an open framework for big data, objects are identified by one or more identifiers, commonly URIs (incl. URL, URN, etc), originating from different identification schemes. Some, but not all, objects, are data, and some, but not all, identifiers permit dereferencing to retrieve this data.
Generally, BOLD objects may be subdivided into :

1. *simple objects*,
2. objects that are decomposable into a *set of BOLD objects*, and
3. objects that – besides being decomposable into sets of objects – are *chains of BOLD objects*, or totally ordered sets of objects, as well.

A BOLD object may participate in multiple decompositions. For chains of BOLD objects one typically encounters order-preserving XML/HTML/PDF data structures. Unordered sets of BOLD objects, when not embedded in XML/HTML/PDF data structures, are preferably structured in RDF or JSON.
Labeled links between BOLD objects are *(subject predicate object)* triples. Triples are simple BOLD objects. These may be characterized where appropriate by

1. a subset R of the possible subjects, predicates, and objects (product S X P X O),
2. a set of *edges* with predicate p E(p) = { (s,o): (s, p, o) in R}, or

3. a set of *features* of a subject *s*, F(s) = { (p,o): (s, P, o) in *R*}.

RDF and JSON data has a standard interpretation as *(subject predicate object)* triples, forming a graph. Graphs are the raw material for application of network analysis techniques, and play an important role in BOLD enrichment. Graphs are moreover BOLD objects, and decomposable into a set of BOLD objects.

2 BOLD objects: social networks

As the producers and users of big open legal data, persons and groups, and links between persons and groups, play a central role in OpenLaws.eu. BOLD enrichment starts with social networks.

Persons come together in groups, shared memberships of a group can be interpreted as links between persons and vice versa, and shared members can be interpreted as links between groups and vice versa, following a social theory attributed mainly to Breiger [4]. Persons may moreover play distinct roles[1] that can be interpreted as group memberships and vice versa. Groups and roles may be directly expressed by BOLD user communities, as a creative act, and they may be discovered through network analysis techniques by looking at their activities as producers and users of BOLD.

BOLD agents, regardless of whether we are dealing with persons, associated with user accounts, or groupings of persons, are therefore decomposable into sets of agents, and persons may decide to which agents productions and uses are associated, and may claim the power to constrain membership or visibility of specific agents.

2.1 Challenges in managing social networks

The OpenLaws.eu project will deal with the following challenges in managing user communities:

1. Permitting user communities to freely organize themselves, to freely share information with the agents of *their* choice, excluding others, and to freely exclude information from view that they deem irrelevant or of low quality;
2. On the other hand, providing incentives to share information that they produce as widely as possible, as implied by the BOLD concept;
3. And finally, determining the relevance of information for specific agents, addressing both the general quality of information (or the confidence in the skills of its producer), and specific agent information needs.

While challenge 2 above is arguably the key objective of the project, challenge 1 is important to the business case for the framework[2], and of central importance to addressing challenge 3, safeguarding the quality of BOLD in the long run.

3 BOLD objects: networks of legal data

Legal data, narrowly conceived, consists of two types of BOLD objects:

---

[1] E.g. a legal scholar specialized in insolvency may at the same time be a part time judge in a cantonal court mainly dealing small claims, and has distinct information needs within these distinct capacities, manifested in his production and use of information.
[2] E.g. for protecting the privacy of the individual practitioner of law and his clients, to allow administrative organizations to recreate organizational and information structures in the framework, and to allow legal publishers to find niches in the framework for selling information and services.

1. structured texts, hierarchically decomposable into chains of text fragments, and
2. metadata about texts and text fragments;
   a. labeled links between texts fragments, and
   b. arbitrarily complex features of texts and text fragments.

3.1 Structural decompositions

Structured texts may require alternative structural decompositions:

1. text structured into chapters and articles may for instance in rare cases require an alternative decomposition into pages, with footnotes;
2. more importantly, in the annotation of individual sentences with markup, annotators may come to alternative structural decompositions of a sentence, depending on purpose of the annotation. It is technically possible to replace sentence annotations with metadata[3], but such solutions are less robust, more complex, and restrict reuse of the decomposition itself by other annotators.

3.2 Bibliographic identity

Bibliographic convention [5] is to distinguish legal texts and text fragments on at least four levels, as distinguished by MetaLex [2][3]:

1. On the *item level* legal texts and text fragments can be dereferenced by identifier and copied, resulting in a new item;
2. On the *manifestation level* any change to the data produces a new manifestation, including a change of data format, annotation of structure, or the embedding of metadata;
3. On the *expression level* only a change of the text by its author produces a new expression;
4. On the *work level* a text is identified by the details of its publication: as long as the title, author, and publication date remain the same, expressions are versions of the same work.

Enrichment consists of 1) creating alternative manifestations of an expression, or 2) adding metadata about an expression or work. One work may be expressed multiple times. An expression may have many manifestations. A manifestation may be copied to many items.

Versioning

For regulatory text, the distinction between works and expressions is of critical importance, because the text is typically changed over time. Most works are decomposable into a single versioning chain of expressions. In some cases (retroactive annulment of changes and unforeseen changes to scheduled changes in the future) the versioning chain may change over time retroactively, resulting in alternative chains of versions of a text.

Languages

Many works are moreover available in alternative language variants. Support of this is obviously an important requirement in the EU.

---

[3] For instance using xpaths/xpointers.

References

Conventionally, regulatory text refers to other regulatory text on the work level: which version should be used is left to the reader (see Figure 1). Some call this a *dynamic* reference. A court decision refers to a specific version of a regulatory text on the expression level. Some call this a *static* reference. Any other text that refers to legislation by default refers to a specific version, unless the text is under editorial control and guaranteed to be up to date with the text it refers to. Obviously, texts may discuss an old version, compare an old version to a new version, compare two language variants of a version, or (very frequently) discuss an anticipated version of a regulatory text[4].
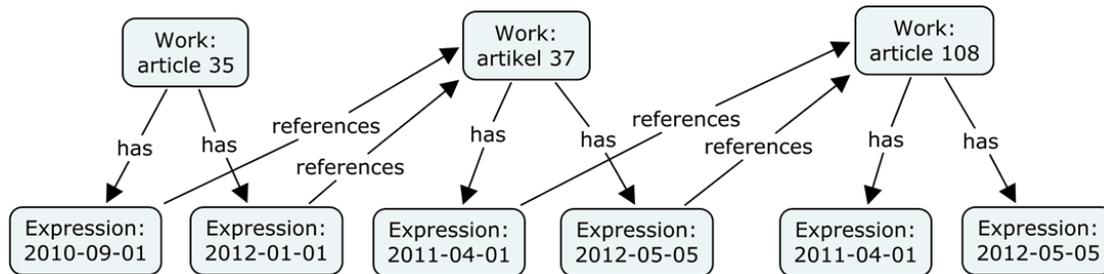

Figure 1: Example of dynamic references in regulatory text. The work ´article 35´ has two expressions; both refer to the work level of article 37. The actual references can only occur in a manifestation of an expression (e.g. an XML file) and be seen by a user in a physical item on for instance her computer.

Mixed content and quoting

Legal text is often quoted, in modifying legislation, in court decisions, papers and books, etc. Quoting is an alternative way of referencing information, and the quoted text fragment is both a part of the quoted and of the quoting document. It is moreover a potential source of interesting and innovative manifestations of text fragments.

3.3   Folders, shopping carts, and shopping lists

Users looking for legal information require some way to select and keep it. One may think of this in terms of the shopping cart metaphor.[5] By storing the contents of a shopping cart in a folder the user enriches the text fragments in the shopping cart with features. The folder is a set based on shared features.

Alternatively, the user may use the shopping cart as a basis for producing a new text, quoting the selected text fragments. In a new text the fragments are ordered in a chain; the shopping list, whose order may be changed, is a more appropriate metaphor for this use case.

Events and other mediating objects

The ultimate purpose of enrichment is to uncover the contexts in which legal data is used and produced. To accurately describe these, user communities should be able to freely introduce entities such as events, business processes, services, logical rules etc. as long as these can be identified by URI. The use case for event-based metadata was discussed in detail in [1].

---

[4] The news value of a discussion of a legal rule is highest well before the new rule goes into actual effect. The importance of anticipation of changes in the law should not be underestimated!
[5] Note that does not imply that users have to pay for the information thus gathered.

Audio and video data

Since OpenLaws.eu will develop app-based support for BOLD, it makes sense to consider audio and video-based legal data. On a smart phone, an audio recording is easier to make than a text. Disadvantage of this mode of annotation is its lack of accessibility for most automated enrichment tools, although it may still play a useful role for network analysis.

3.4   MetaLex conformance

OpenLaws will be based on conformance with the MetaLex standard for legislative documents. The design principles for BOLD objects identified thus far are therefore inspired to a large extent by MetaLex.

OpenLaws does not cover just legislation, but other sources of legal information as well, besides user-managed shopping lists and folders. These documents and folders will 1) refer to legislation, meaning that the references should meet the requirements, and 2) literally quote from (or include) legislation, meaning that they embed alternative manifestations of a part (a article, a sentence, etc) of the legislation.

Implied design requirements

In this section we list MetaLex-based requirements briefly, for convenience of the reader. Full requirements are found in the specification[6]. The main design principles of MetaLex [2][3] are the following:

1. Legislative documents, and their parts, can be individuated on the work, expression, manifestation, and item levels of abstraction.
2. The HTML or XML structure is a manifestation of a document and its parts. The structure of documents can be described without ambiguity using a limited number of content models.
3. Work, expression, and manifestation, and their parts, should have a unique name that is an IRI reference.
4. These names should meet the requirements of some naming convention. A transparent (meaningful) name may be interpreted as a set of identifying metadata, and vice versa, a set of identifying metadata may be associated to an opaque (meaningless) name instead.
5. Metadata in general is about the document as a work, as an expression, or as a manifestation. It uses the right IRI reference as a subject.
6. Metadata can be interpreted as RDF triples.
7. Most legislative metadata describes events that happened (or will happen) with the document (as a work, expression, or manifestation)[7].
8. References to legislative documents made by authors usually refer to the work (dynamic reference), or to the expression (static reference), and not to the manifestation or item level. A correct technical reference is not the same as a direct hyperlink.

Note that it is possible to make XHTML documents conform to this standard. Another

---

[6] ftp://ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA15710-2010-Metalex2.pdf

[7] Example of points 5 and 7: a last-modified property with a date means different things depending on whether it applies to 1) the expression (the actual text was modified by its author, the legislator) or 2) the manifestation (the XML or HTML markup was modified by an editor, but the text remained the same). Moreover, the date is not a direct property of the document, but of an event (a modification) that either 1) happened to the work, and resulted in the expression, or 2) happened to the expression, and resulted in the manifestation.

important source of possible input is Akoma Ntoso XML, which is known to conform to MetaLex requirements.

Transforming existing text

To transform existing documents from other corpuses, consider:

1. The mapping to MetaLex content models
2. Decide on a naming convention. Do existing document identifiers and metadata give enough information to distinguish and name works, expressions, and manifestations? How much of the version history of the documents is available? Are the documents multi-version manifestations, that should be cut up and replicated into multiple documents?
3. Reinterpret embedded metadata as event descriptions, described by RDF triples.
4. Resolve references, either by looking at link metadata, or IRI references, or the text of the reference itself. Do existing references make clear whether work, expression, or manifestation are the intended targets of a reference?

Local replication of documents and metadata

1. During the project, documents may be replicated in the OpenLaws infrastructure[8]: although there are good reasons not to replicate data, replication will make development of tools easier.
2. Since the repository(y/ies) should be up to date, incremental, non-destructive updates should be made automatically in that case[9].
3. Metadata will be stored/replicated in a dedicated repository, instead of depending on it being embedded in the (manifestations of) documents, to make development of tools easier.
4. Embedded metadata in documents viewed by users duplicating metadata in the repository may exist, but is not directly accessed by tools.
5. Users should be able to export documents from the OpenLaws infrastructure for purposes of a) printing and b) archiving self-contained and self-describing documents. For this second purpose, retaining original metadata, or even adding inserting newly created metadata from the repository or user folders, is important.
6. Note that user-created metadata should also be considered to be stored in a user-created folder, however, and user-created metadata marked as private is not shared with the repository.

Meeting requirement 2 for all corpuses of text included in the OpenLaws infrastructure may not be realistically feasible in the project. Within the project, it may not be realistically feasible to embed a concise metadata description of a document in the document based on the entire metadata repository[10], as implied by requirement 5.

4 BOLD objects: overview

The figure below presents a taxonomy of BOLD objects. For bibliographic objects

---

[8] similar to the approach of example server http://doc.metalex.eu
[9] The source code of the updating script of example server http://doc.metalex.eu is freely available as an example.
[10] Depending on the structure of the repository and the logical inference involved, this may be a prohibitively hard problem.

and text fragments, additional views can be found in the MetaLex OWL schema. Note that texts are manifestation-level entities, although not all texts may be considered true bibliographic objects. Texts that are not part of a full document are not bibliographic objects.

The enrichment tools do not embed metadata into documents, and do not necessarily take account of any metadata present in documents. Metadata is embedded in folders made by users and automated enrichment tools. The shopping list is a mediating object that permits user editing before it is saved as a folder in another folder. As the metaphor implies, only one shopping list is active while a user is browsing for relevant BOLD objects.

Dependent bibliographic identity may for instance occur for treaties and associated authorized translations and protocols, acts of parliament with associated explanatory memorandum or authorized translations, etc.

The other object types are self-explanatory.
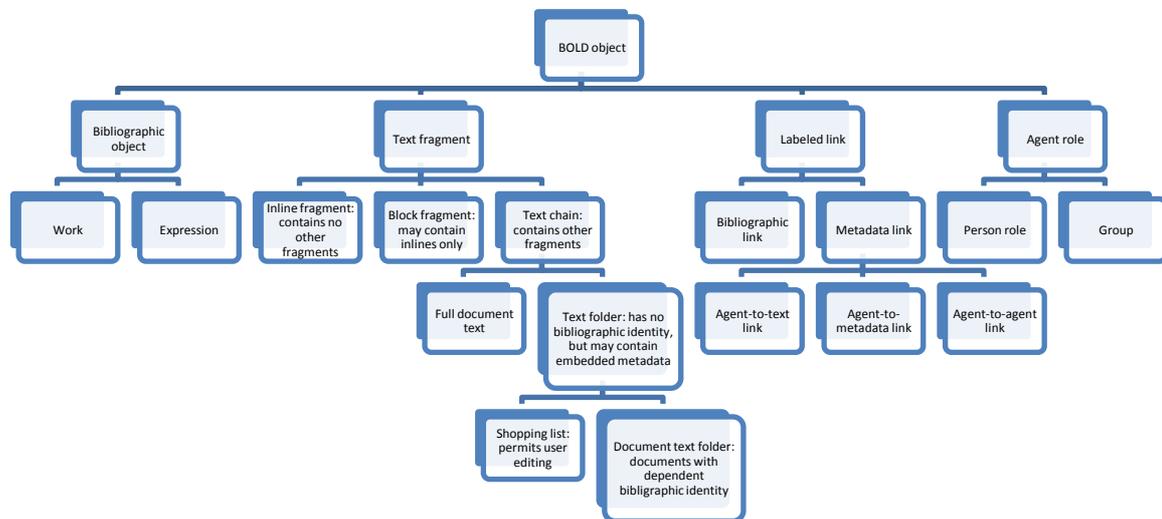
5 Accessing documents and texts

5.1 Requirements for text fragments

The text fragment is the fundamental unit of the user interface: (named) containers, like article 1, paragraph-sized blocks, or sentences.

Views

1. Displaying a text fragment, in a resizable view, with an appropriate layout.
2. Displaying a text fragment chopped to a specific character length, or until the fragment meets certain information requirements.
3. Displaying a globally correct citation of a text fragment, on the expression level, including date of last revision or publication and a title[11].

---

[11] For instance *LJN AI5638* is a correct case law identifier, but not informative to the reader. *Gerechtshof 's-Gravenhage 4 september 2003, LJN AI5638 (Scientology)* is perfect.

4. Displaying a *locally* correct citation of a text fragment, relative to the text of which it is part, not repeating elements already obvious[12].
5. Drawing attention to inline annotations to permit manipulations.

Manipulations

1. Folding or unfolding text fragments to show more or less information.
2. Selecting a text fragment.
3. Adding the text fragment to the shopping cart.
4. Selecting an inline annotation in a selected text fragment, and performing the appropriate action.
5. Accessing alternative known manifestations of the text fragment.
6. Accessing the previous and next expression-level version of the text fragment.
7. Accessing alternative language variants of a text fragment.
8. Accessing closely related texts[13].

Process and provenance aspects

1. Correct citation style depends on the type of document. To access the information for a globally correct citation one should have access to the entire text. In some cases the ability to select alternative styles may be required[14]. Coverage of all relevant styles is not realistically feasible in the project.
2. Appropriate layout may depend on the type of document.
3. Known manifestations of a text fragment include those that only exist in a user folder, or as quoted content of another text.

5.2   Requirements for text fragments in a shopping list or shopping cart

In addition to all other requirements for text fragments.

Manipulations

1. Creating inline annotations[15] in a selected text fragment[16].
2. Adding features to the text fragment.

Process and provenance aspects

1. As required by basic XML/HTML, annotations in a *single manifestation* of a text fragment should not overlap.

5.3   Requirements for chains of text fragments

Applies to any text fragment that can be broken into a chain of text fragments, including full texts and shopping lists. All requirements of text fragments apply, except that the chain of text fragments is not chopped to a specific length, but is applied to its

---

[12] For instance *article 1*.

[13] Incoming and outgoing references, shared features, texts that quote it, etc.

[14] The ECLI for instance competes with state-specific conventions like the example in footnote 5. For academic article and book citation styles existing solutions may be available.

[15] This requirement is very important, but may be prohibitively hard to realize in the project.

[16] Because a new manifestation of the text fragment is created that does not replace the one which was edited. If we do not replicate original documents in OpenLaws, this new manifestation may be lost or inaccessible for other users.

parts.

Views

1. Chopping (long) text fragments to a specific character length, or until the fragment meets certain information requirements, to show more of the text in the screen.

5.4   Requirements for shopping lists and shopping carts

Shopping lists are chains of text fragments that can be edited, to create a new text[17]. Other requirements apply. A shopping cart is a shopping list that has not been explicitly ordered or edited (yet).

Manipulations

4. Reordering text fragments in the list.
5. Adding features to the shopping list.
6. Writing a new text fragment and inserting it in the place of choice in the shopping list.
7. In an app platform for smart phones: recording audio or video and inserting it in the place of choice in the shopping list.

6 Enrichment tools

6.1   Users working with shopping lists

Views

1. Are ordered sets, and may be reordered: think of them as newly created hypertexts.
2. May include metadata (triples), text fragments and entire documents, and audio and video recordings.
3. Any texts or text fragments are included in the shopping list qua work, expression or manifestation.

Manipulations

1. The user drops resources into the shopping list while browsing, or creates a text or audio or video recording himself.
2. May be reordered.
3. Allows annotation of text fragments with XML tags (qua manifestation), resulting in a new manifestation of that text fragment that replaces the original locally.
4. Can be saved as a user-managed folder.

Process and provenance aspects

1. It is possible to treat the shopping list and user-managed folder as a MetaLex XML file.
2. Membership by reference (href), by inclusion (src) should be distinguished.

---

[17] The interaction model we have in mind is that of for instance *Storify*.

6.2   Users working with user-defined folders

Views

1. The folders view models file explorer widgets.
2. Folders may be associated with descriptive metadata.

Manipulations

1. The user-defined folder is created 1) as a new folder or 2) by saving a shopping list in a parent folder and giving it a title, and descriptive metadata.
2. Folders may be made (in)visible to other groups.

6.3   Users managing and groups and roles

Views

1. Hierarchical view like the folder view

Manipulations

1. Creating groups and roles
2. Adding yourself (role) to an open access group
3. Requesting (role) invitation to a group
4. Inviting others by role to a group you are member of
5. Switching active role for creating a shopping list

6.4   Search and recommendations

Views

1. Corpuses are hierarchically laid out as folders, and filters can be applied to them. Visible user-defined folders can be searched as well.
2. Search results are displayed as chains of text fragments.

Manipulations

1. Search a corpus with keywords, optionally a reference date (expression level or work level search), and a granularity setting (size/type of the text fragment in which the keywords should occur).
2. Search results can be saved as a user-defined folder, and the search can be rerun on the folder.
3. Recommendation of search results: text fragments in search results can be reordered, and hidden from view by the user.
4. Recommendation of user-defined folders.
5. Recommendation of users/roles/groups.

Process and provenance aspects

1. Should keep a full log of recommendation decisions, including the decision-making users/roles. This is important for improving network analysis over time.

6.5 Network analysis functions

Views

1. For any text fragment, alternative expressions and manifestations of that text fragment;
2. All incoming and outgoing references, and co-referenced text fragments[18];
3. All metadata that has the text fragment as subject or object;
4. All folders of which the text fragment is a member; and
5. Any list of results should be ranked based on recommendation.

Process and provenance aspects

1. The OpenLaws infrastructure should make all relevant network data available for daily download in packages.
2. The frequency with which network analyses are run should be clear to the user.
3. All network data and recommendations have an author, being the author of the document from which data was taken, an OpenLaws user, or a network analysis algorithm. Determining the confidence in authors always has a high priority.
4. One recommendation process randomly assigns recommendations, and monitors changes, for validation purposes.

6.6 The OpenLaws.eu production pipeline

Although the OpenLaws demonstrator will use a single server instance, the concept of an open infrastructure implies a decentralized infrastructure, not managed by a single information owner. Synchronization of legal information is therefore not guaranteed for the future, and network analysis functions should not depend on the assumption that all legal information is always synchronized. A certain *core subset* of metadata, extracted from documents as they are passed through the pipeline, is however shared between all instances of OpenLaws.eu.

Process and provenance aspects

1. Certain network analysis functions belong to the core functionality of the OpenLaws.eu functionality: all instances, if using the same version of all components, will extract the same metadata from a document.

2. The operative principle for core functionality is (test-retest and inter-rater) reliability. If there are differences of opinion about whether a function meets this standard, it does not.

3. To the core functions belong: inverted indices for keyword search, and derivative metadata, publication metadata, and metadata on the resolution of references between documents.

---

[18] Text fragments that are referenced by the text fragment that references the focus text fragment.

References

[1] A. Boer. Using event descriptions for metadata about legal documents. In: R. Winkels and E. Francesconi, editors, *Electronic Proceedings of the Workshop on Standards for Legislative XML, in conjunction with Jurix 2007*, 2007.

[2] A. Boer and T. van Engers. A MetaLex and metadata primer: Concepts, use, and implementation. In Legislative XML for the Semantic Web, pages 131–149. Springer, 2011.

[3] A. Boer, R. Winkels and F. Vitali. MetaLex XML and the Legal Knowledge Interchange Format. *Computable Models of the Law*. Lecture Notes in Artificial Intelligence 4884, pp. 21-41, Springer Verlag, Berlin, 2008.

[4] R.L. Breiger. The duality of persons and groups. In: B. Wellman, S. Berkowitz (Eds.), Social Structures: Network approach, Cambridge Univ. Press, Cambridge (1988), pp. 83–98.

[5] K. G. Saur. Functional requirements for bibliographic records. *UBCIM Publications - IFLA Section on Cataloguing*, 19, 1998.