Promoting Scientific Creativity by Utilising Web-based Research Objects

Integrated Project (IP)

FP7-ICT-2013-10. Information and Communication Technologies

Grant Agreement Number 611383

José Luis Redondo García, Carlos Badenes Olmedo, Óscar Corcho
Universidad Politécnica de Madrid

Deliverable D5.6

Techniques of Personalised Recommendations of ROs with Report

Grant agreement no: 611383

Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	**X**
CO	Confidential, only for members of the consortium (including the Commission Services	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	Dr Inventor
Project Full,Name:	Promoting Scientific Creativity by Utilising Web-based Research Objects
Deliverable No.:	D5.6
Document name:	Techniques of Personalised Recommendations of ROs with Report
Nature (R, P, D, O): [1]	Report
Dissemination Level (PU, PP, RE, CO): [2]	RE
Version:	v1.0
Actual Submission,Date:	21/11/2016
Internal Reviewer:	Diarmuid O'Donoghue
Editor: Institution: E-Mail:	José Luis Redondo García Universidad Politécnica de Madrid (UPM), Madrid, Spain jlredondo@fi.upm.es

ABSTRACT:

This deliverable describes the different knowledge discovery and recommendation mechanisms implemented in the DRInventor Platform. The objective of these techniques is to provide users with the relevant knowledge available in big corpora of documents in an automatic and timely manner. This way consumers in general and scientists in particular can obtain pertinent suggestions about which resources or specific fragments inside those resources they should be reading next, hence maximising the relevance of the information consumed while minimising the efforts spent on identifying them.

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

KEYWORDS LIST:

Research Object, Recommendation, Knowledge Discovery, Topic Modelling

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013.8.1) under grant agreement no 611383.

The author is solely responsible for its content, it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

MODIFICATION CONTROL				
Version	**Date**	**Status**	**Author**	**Comments**
0.1	06/08/2016	Draft	José Luis Redondo García	Initial version of 5.6
0.2	06/08/2016	Draft	José Luis Redondo García	Description of RO repository and topic modelling
0.3	07/09/2016	Draft	José Luis Redondo García	Adding recommendation methods
0.4	16/09/2016	Draft	Carlos Badenes	Review of topic-based similarity measure
0.5	03/10/2016	Draft	Óscar Corcho	Rephrasing and quality check
0.6	10/10/2016	Draft	José Luis Redondo García	Reviewing Oscar's suggestions
0.7	04/11/2016	Draft	José Luis Redondo García	Applying Internal Review comments
1.0	10/11/2016	Final	José Luis Redondo García	Minor changes and generating final version

Executive Summary

This deliverable describes the different knowledge discovery and recommendation mechanisms implemented in the DRInventor Platform. The objective of these techniques is to provide users with the relevant knowledge available in big corpora of documents in an automatic and timely manner. This way consumers in general and scientists in particular can obtain pertinent suggestions about which resources or specific fragments inside those resources they should be reading next, hence maximising the relevance of the information consumed while minimising the efforts spent on identifying them.

This document also provide examples and use cases of recommendation operations in the DRInventor Platform where those different methods are exploited, giving an idea of the promising potential they hold. We will also show how those functionalities are exposed through an API, so that they can be programatically accessed. Some of them are also available in a more human-friendly and interactive way via the DRInventor Dashboard.

Summary of Novelty

The main contributions implemented in DRInventor Platform in order to offer innovative recommendation operations compared with state-of-the-art approaches are:

1. A Tailored-made Repository of Research Objects. DRInventor is able to ingest and index research resources from external sources at different levels of granularity: from the entire documents, to their individual items, parts or even individual words contained in them. On top of those resources DRInventor attaches different annotations that further describe the instances and give support to different operations leveraging on them. This model provides a standard way of representing research documents, and is flexible enough to give support to a great variety of analysis techniques bringing value to the information stored in it.

2. Advanced Exploitation of Probabilistic Topic Modelling and Ontology Learning Techniques. Those methods describe the indexed resources in a more cognitive way thus automatically capturing aspects that a human annotator would consider as important: the subjects a document is talking about, or the potential of certain terms for representing an important concept in the underlying domain. Those techniques make possible to cluster the resources in research areas, or to establish relations between similar resources so that we can efficiently traverse the knowledge inside the corpus.

Table of Content

0 Deliverable Structure 7

1 Introduction 8

2 The Research Objects Repository in a Nutshell 10

2.1 General Overview of the RO's Repository . . . 10
2.2 Managing the RO's Collection . . . 12
2.3 Resources inside the RO's Repository . . . 14
2.4 Annotations inside the RO's Repository . . . 17
2.4.1 The Analysis class in DRInventor . . . 17
2.4.2 List of Supported Annotations . . . 18
2.5 Exploring Resources and Annotations in the DRInventor Platform . . . 19
2.5.1 The SIGGRAPH Corpus via the DRInventor Dashboard . . . 20
2.5.2 The SIGGRAPH Corpus via the REST API . . . 22

3 Probabilistic Topic Modelling for Generating RO's Relations 28

3.1 Introduction to Probabilistic Topic Modelling . . . 28
3.1.1 Latent Dirichlet Allocation . . . 29
3.1.2 Topic Visualisation in DRInventor . . . 31
3.2 Configuring LDA using an Evolutionary Multi-objective Optimisation . . . 34
3.2.1 Multi-Objective Evolutionary Approach . . . 34
3.2.2 Experiment: Optimising LDA for a Corpus of Research Objects . . . 36
3.3 Topic-Based Resource Similarity . . . 39
3.3.1 Content-based Similarity . . . 39
3.3.2 Context-based Similarity . . . 41
3.3.3 Similarity Measure sim_D over a RO Corpus: Experiments and Results . . 44
3.3.4 Materialising Relations between Resources based on their Similarity . . . 47

4 Recommendation Functionalities in DRInventor Platform 49

4.1 Get Relevant Resources given a Document . . . 49
4.2 Get Relevant Resources given a Text Excerpt (Filters) . . . 51
4.3 Get a Path between two Documents . . . 54
4.4 Cluster Documents according to Topics . . . 61
4.5 Other Advanced Operations in the DRInventor Platform . . . 61
4.5.1 Generating Knowledge Routes . . . 62
4.5.2 Towards a Linked-Research . . . 64
4.5.3 Defining an Optimal Paper Review Process . . . 65

5 Conclusions 66

6 Future Work 67

6.1 Short Term Developments in DRInventor Platform 67

6.2 Boosting the Performance of Implemented Techniques 68

6.3 Evaluating the Implemented Techniques 68

6.4 Implement the Notion of Users Using the the Platform 69

0 Deliverable Structure

In Section 2 we describe the repository of Research Objects that was previously introduced in Deliverable 5.4. In particular, we specify how the different papers are indexed into the platform (items, parts, documents, words...), and the different annotations that are generated by relying on them (topics, terms, clusters...).

Section 3 includes the key contributions of this deliverable by elaborating in two main pillars: 1) how the topics are generated out of the indexed research objects and which are the algorithms used in DRInventor to generate them, and 2) how those topics are used in order to find connections between research resources, therefore providing interesting features grounded on the idea of discovering and navigating the knowledge available in the corpus.

In Section 4 the aforementioned techniques are put into practice through different example features that showcase the powerfulness of DRInventor when proposing new scientific documents or their sub-parts. In particular we propose four different use cases that are exposed both via de REST API[3] and the public prototype[4].

Section 5 wraps up the main contributions of this deliverable and gives a final overview about the possibilities offered by DRInventor for the recommendation of RO's. Finally in Section 6 we sketch the future lines of the recommendation capabilities described in this document, making special emphasis in the great potential of the platform for improving the way scientific can discover new relevant knowledge among the huge quantity of new research papers that are published everyday.

[3] http://drinventor.dia.fi.upm.es/api/

[4] http://drinventor.dia.fi.upm.es/

1 Introduction

Given the huge amount of information about any domain that is being produced or captured daily, it becomes crucial to provide mechanisms for automatically discarding all the noisy, non-relevant information and keeping only the data that can bring value for the involved agents (general consumers, experts, companies, investors...)

In the context of the scientific domain targeted by DRInventor, the personalised recommendation of research objects based on their content is a key feature for performing a smart selection of relevant resources over very big datasets. From the set of values and different attributes of the RO's and by generating advanced knowledge models about the information they contain (including terms obtained by ontology learning techniques or topics being described) we can bridge across the different relevant pieces of information and allow users to navigate them in a more efficient and powerful way.

It is important to clarify that the research process is a hard, exhaustive and dedicated multi-stage procedure that researchers put lots of efforts into. Each of its stages presents difficulties and challenges that may block the entire process so any automatic assistant providing help on them can make the difference.

To give a deeper view on the complexity of the Research Method, below we will highlight some of its main tasks and constituents. According to [5] the Research Method is an approach to the process of inquiry, in which empirically grounded theory of nature is constructed and verified for increasing the available human knowledge, based on systematic observation, classification and interpretation. It is characterised by objectivity, generality, verifiability and creditability to ensure an unbiased, general and impersonal study [24]. This process consists of a series of steps or actions that are important to execute a specific research in an effective way, such as: (1) *define a research problem*, (2) *review literature*, (3) *write a hypothesis*, (4) *design the research*, (5) *collect data*, (6) *analyze the collected data* and (7) *interpret results and report* [26].

While each of these steps inside the Research Method is important, the first one is crucial because it will determine how we reach the rest of steps and ultimately the final goal. Every research effort should address a unique issue and build upon previous research and scientifically accepted fundamentals. Besides, some important aspects need to be considered in order to maximise the outcome of the process: the **interestingness** or determining how relevant the investigation is for the researcher and the community, the **Magnitude** so we make sure that it is manageable and we have enough time and resources to solve it, the required **Expertise** for the scientists involved to make sure they can carry out the research, and the **Availability of data** for making sure that enough data to conduct the investigation is available. When the problem is clearly identified, it is time to *write a set of hypotheses* that need to be proved with new experiments and observations. Usually it is the result of a process of inductive reasoning from observations to create a testable, falsifiable and realistic statement. Finally, the researcher has to present the ***interpretation of results and the report*** derived from this study in a structured and logical manner following a systematic, chronological or psychological order. The most important thing is to prune out irrelevant information and findings.

All those considerations and requirements can be relaxed and the results and objetives better achieved if we try to help researchers during this complicated procedure. This is the very ambitions and ultimate challenge of DRInventor, mainly through the implementation of advanced operations like recommendations. Therefore the objective is to emulate what a human scientific assistant would do: to provide researchers with useful information for each stage of the investigation process. The different contributions described in this document represent a sound balance between scientific creative operations involving new knowledge discovery features, and technical implementation using innovative technologies in information extraction, document summarisation and Semantic Web.

2 The Research Objects Repository in a Nutshell

This section describes the Research Objects repository by making emphasis on the different resources that are available on it and how they relate to each other. This gives an overview of the possibilities that DRInventor is able to offer and how it can give support to different operations such as recommendations (for more information see Section 3).

For the rest of the deliverable, we assume that the harvesting and indexing process of research documents described in section 3.4.2 and 3.4.3 of the Deliverable 5.4 [4] has been already performed. Hence, all the sources have been accessed and the different research objects downloaded by the *Hoarder* module and decomposed into different logical units and indexed by the *Harvester* in order to make them available in the platform.

DRInventor Platform is powered by the Librairy[5] toolbox, which provides a set of components for easily retrieve, index, and process big collections of research objects. Many of the features offered by DRInventor are exposed also as a module inside the framework so they can be reused by other similar projects trying to deal with huge sets of documents.

There are two main kinds of information units in the repository: the so-called *Resources*, which are directly generated from the ingested research objects and still have a local scope related to the document where they were extracted from, and the *Annotations* which are generated once the resources from a Domain are available in the platform. They were not explicitly available in the original documents but bring a higher potential to perform advanced operations over the corpora, such as recommendations.

2.1 General Overview of the RO's Repository

In this first subsection we present an overview of the different units that together shape up the repository of DRInventor, which will be further described in successive sections below. The system is updated with both gathered and submitted information that is internally modelled as resources. A resource is an abstract representation of the information contained in any of the retrieved research units, which is characterised by the type of content available in the unit and the status.

The main types of resources that are considered in DRInventor, from the most fine/grained to the most general ones, are:

– *Word*: a meaningful element of writing inside a document, formed by a sequence of characters with no blanks.

– *Part*: logical division of a document, based on categories of the research discourse such as abstract, introduction, methods, results, conclusions, etc., including also the types of

[5] `https://github.com/librairy`

rhetorical sentences[6] (i.e. approach, background, challenge, future work or outcomes).

- *Item*: each of the elements that make up a research object (i.e. a document) such as a paper, programming-code, an image, a workflow, and so on.
- *Document*: meta-information retrieved from a research object. A document is composed by a set of items.

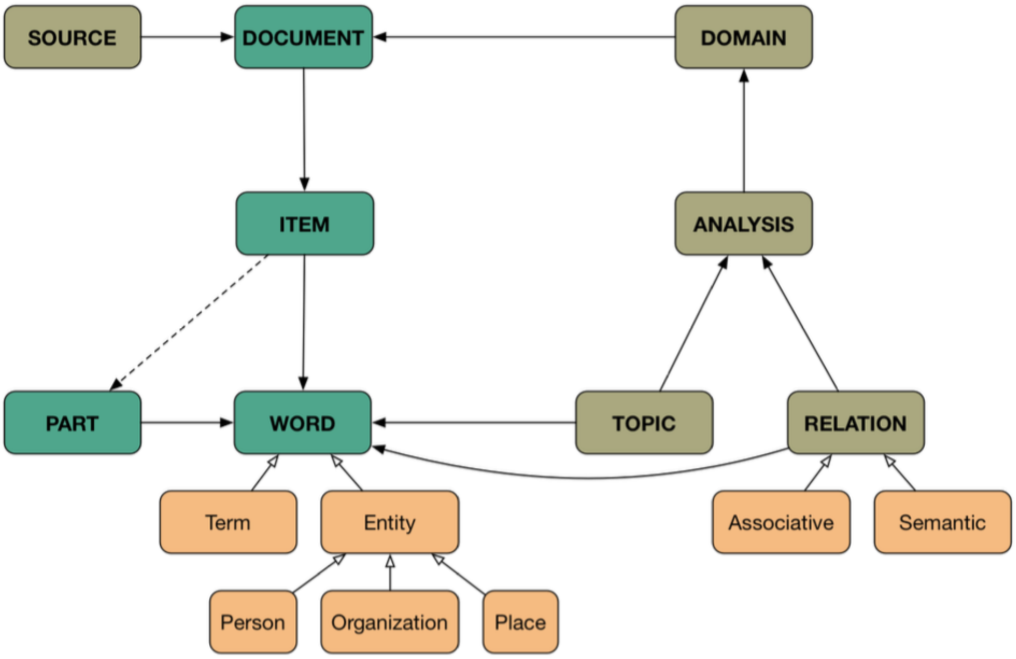


Figure 1: Overview of Resources in DRInventor Platform

The documents and derived resources ingested in the platform are grouped together via two global classes named *Source* and *Domain*:

- The *Source* indicates the repository where the raw research objects were collected from via a link where the platform can look at in order to retrieve them.
- The *Domain* represents the collection of the resources inside DRInventor generated after ingesting the research objects from the repository specified by the Source.

Finally, different annotation algorithms relying on the generated resources are used in order to produce further descriptions about documents and the corpus that were not explicitly available in the original research objects. This information is represented through the following classes:

[6] `http://backingdata.org/dri/library/1.0.5/doc/edu/upf/taln/dri/lib/model/ext/RhetoricalClassENUM.html`

- *Analysis*: it represents the execution of an algorithm over a particular Domain in the platform. It is responsible for the creation of annotations, such as topics and relations.
- *Term*: represents and abstract concepts, such as entities (persons, locations, organisations...) as results of the execution of different Natural Language Processing algorithms.
- *Topic*: helps to materialise the main subjects that the corpus is elaborating on, such as the research areas or trending issues in the scientific domain.
- *Relation*: associative or semantic connection between resources in a domain. They can model for example the high degree of similarity between two Parts belonging to different research objects.

To better illustrate this model through an example, consider that we want to add a new research paper into the system. First, this paper will be materialised as a new document containing information such as title, author(s), publisher or language. Immediately after an instance of an Item is also created, for this particular example including the same title, authors and metadata together with some keywords and attached to the original document. Considering the research object would have contained other material like pictures or even video/audio, they would have been serialised as different items belonging to the same document. Moreover, this item will be associated to several Parts, each of them grouping sentences by rhetorical class (e.g. approach, background, challenge, future work and outcome) or by section (e.g abstract, introduction). Those parts will be in turn composed by Words. Finally through the different Analysis performed in the platform, the initial set of resources will be extended with more annotations representing Topics, Relations, and Terms.

2.2 Managing the RO's Collection

As introduced before, in the repository we leverage on two different classes to manage collection of resources at a global scope:

Source A source is a repository of research objects. It contains the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `sources/de305d54-75b4-431b-adb2-eb6b9e546014`
- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF86014[7].
- name: a label associated to the resource.
- description: additional information about it.

[7] http://www.iso.org/iso/home/standards/iso8601.htm

- url: the Uniform Resource Locator of the repository.

- protocol: defines how the digital content is published.

A source may contain zero or more references to documents and a document may have one or more references to sources, i.e. the same document can be available in more than one source. The underlying repository a source is pointing to may be static or dynamic:

- Static: the repository will not change along time. So, once processed, new information will never be collected from it again. It can be a single file (e.g. `http://world.std.com/~rjs/indinf56.pdf`) or a closed time-based expression for an open archive service (e.g. `http://www.worldsciencepublisher.org/journals/index.php/AASS/oai?from=2012-01-01T00:00:00`).

- Dynamic: the repository may have new documents being added the future, such as an open archive publisher (e.g. `http://oa.upm.es/perl/oai2`), a RSS feeder (e.g. `http://rss.slashdot.org/Slashdot/slashdot`) , a remote folder (e.g. //192.168.5.125/Public/pap or even a web page (e.g. `https://en.wikipedia.org/wiki/Artificial_Intelligence`). This type of resources will be continuously polled by the hoarder module.

Currently, The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)[8] and Really Simple Sindication (RSS)[9] are the protocols supported by our system. Future integrations will be done to allow sources such as Elsevier API[10], Research Gate[11] or Mendeley[12].

Domain A domain is a logical grouping of documents. It defines the workspace for the modeller and the learner module. By default, all sources have an associated domain with their documents. It contains the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `domains/de305d54-75b4-431b-adb2-eb6b9e546014`.

- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF8601.

- name: a label associated to the resource.

- description: additional information about it.

Furthermore, a domain can contain zero or more references to documents and a document may be referenced by one or more domains.

[8] `http://www.openarchives.org`

[9] `http://www.rssboard.org/rssFspecification`

[10] `http://dev.elsevier.com/`

[11] `https://www.researchgate.net/topic/api`

[12] `http://dev.mendeley.com`

2.3 Resources inside the RO's Repository

In this section we specify in more deep the different resources that are generated when a repository of research object specified by a Source class is ingested by the DRInventor Hoader and Harvester modules and indexed according to the internal schema.

Document A document contains all the meta-information associated to a research object. According to the Open Archives Initiative for Object Reuse and Exchange[13] (OAIFORE) manifest for research objects and the Dublin Core Metadata Annotation[14], it could include:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g documents/de305d54-75b4-431b-adb2-eb6b9e546014
- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- publishedOn: the time the resource was published.
- publishedBy: an entity responsible for making the document available. It can be a person, an organisation or a service. It may be different from the entity that conceptually formed the resource (e.g. wrote the document), which should be recorded as authoredBy. This entity should be identified by a valid Uniform Resource Identificator (URI), e.g. WebId[15], orcid[16] or internal URI.
- authoredOn: the time the research was conceptually formed. The author time should be present if different from publishedOn. It must be a formatted timestamp following ISOF8601.
- authoredBy: an entity primarily responsible for making the content of the document. It may be a list to indicate multiple authors. Each of them identified by a valid URI following WebId or orcid schemas, for example.
- retrievedFrom: a URI identifying the source from which the document was derived. This property should be accompanied with retrievedOn.
- retrievedOn: the time the document was retrieved on. If this property is present, then retrievedFrom must also be present. It must be a formatted timestamp following ISOF8601.
- format: the physical or digital manifestation of the resource. Typically, it includes the media type, i.e. the IANA[17] code of the document.

[13] http://www.openarchives.org/ore/1.0/toc.html

[14] http://dublincore.org/documents/1999/07/02/dces

[15] http://www.w3.org/wiki/WebID

[16] http://orcid.org

[17] http://www.iana.org/assignments/mediaFtypes/mediaFtypes.xhtml

- language: the language(s) in which the document is specified. It is defined by RFCF1766[18] which includes a two-letter language code followed, optionally, by a two-letter country code.
- title: a name given to the document. It is a name by which the document is formally known.
- subject: keywords, key phrases or classification codes annotated by the authors that describe a topic of the resource.
- description: an account of the content of the document. It may include but is not limited to an abstract, or a free-text account of the content.
- rights: information about rights held in and over the document.

Furthermore, a document may contain zero or more items. In turn, an item can belong to one or more documents. Since *epnoi*[19] can also discover analogies among documents, a document may contain zero or more references to other documents.

Item An item is each of the elements that make up a document (i.e. the files bundled in a research object). It may be a paper, programming code, an image, a workflow or any other media content. It includes the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `items/de305d54-75b4-431b-adb2-eb6b9e546014`.
- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- authoredOn: the time the research was conceptually formed. The author time should be present if different from publishedOn. It must be a formatted timestamp following ISOF8601.
- authoredBy: an entity primarily responsible for making the content of the document. It may be a list to indicate multiple authors. Each of them identified by a valid URI, e.g. WebId, orcid.
- format: the physical or digital manifestation of the resource. It includes the media-type, i.e. the IANA code.
- language: the language(s) in which it is specified. It is defined by RFCF1766 which includes a two-letter Language code followed, optionally, by a twoletter country code.

[18] `http://www.ietf.org/rfc/rfc1766.txt`

[19] `https://github.com/epnoi/epnoi`

- title: a name given to the item. It is a name by which the item is formally known.
- subject: keywords, key phrases or classification codes annotated by the authors that describe a topic of the resource.
- description: an account of the content of the document. It may include but is not limited to an abstract, or a freetext account of the content.
- url: path to the file. e.g. pdf file path or png file path.
- content: textual annotation about the file. When it is a paper, it contains the raw-text of the paper. When it is an image, it contains the textual description of the image.

Furthermore, an item may contain zero or more parts and one or more words. In turn, a part only belongs to one item, and a word can belong to one or more items. Since *epnoi* can also discover analogies among items, an item may contain zero or more references to other items.

Part A part is a logical section in an item. When an item is a paper, for instance, it will have as parts the rhetorical classes identified in the sentences of its textual content. It contains the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `parts/de305d54-75b4-431b-adb2-eb6b9e546014`.
- creationFtime: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- sense: content-type. It could be a rhetorical class such as background, approach, challenge, future work or outcome; or a section in the text such as introduction, abstract, discussion, conclusion, results or method; or any other label used to classify parts of a text.
- content: text retrieved from the text of the item, sharing the same class in a classification (i.e. contentFtype).

Furthermore, a part can contain one or more words and a word can be referenced by one or more parts. Since *epnoi* can also discover analogies among parts, a part may contain zero or more references to other parts.

Word A word is a term or an entity (i.e. person, organisation or place) or any other meaningful unit contained in a text. It may include linguistic annotations such as lemma, stem and part-of-speech (POS) after a certain analysis is performed over it. It contains the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `words/de305d54-75b4-431b-adb2-eb6b9e546014`
- creationFtime: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- content: the alphanumeric character string.
- lemma: the word which stands at the head of a definition in a dictionary.
- stem: the root of the word.
- pos: (part-of-speech) the syntactic category of the word, after performing a POS analysis over the word.
- type: term or entity.

2.4 Annotations inside the RO's Repository

There is a second kind of element stored in the platform that is not directly generated by serialising the information coming from the Sources. Instead, it is generated once the corresponding Research Objets are retrieved from the repositories, and after or during the serialisation process. Therefore they bring into the table a set of clues about the data represented in the corpus, which were not explicitly available before. DRInventor contains some modules that are in charge of performing those analysis. In the current version, software components like the Modeler (which calls the NLP tools over the text being retrieved), the Learner (that generates the terms of a vocabulary that can fit the information in the corpus) and the Comparator (which finds similarities between the different resources according to the Topics), are working to produce those more precise descriptions that allow us to offer advanced operations to the final users of the platforms. In future development efforts we intend to add new modules or modify the existing ones in order to keep improving the quality of those annotations. The current most important annotations considered by the platform are listed below. For more information about those modules and the way they produce those annotations, check out the Deliverable 5.4 [4].

2.4.1 The Analysis class in DRInventor

An analysis is a study carried out over the documents contained in a domain, in order to produce further clues about the corpus that help us to browse or discover the data in it. In the current implementation, the analysis is mainly focused on items but may also include parts for deeper analyses. Its main purpose is to discover topics and relations in the domain and calculate the similarity values among its documents. It can contain the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must contain a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `analysis/de305d54-75b4-431b-adb2-eb6b9e546014`.
- creationFtime: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- description: details about the algorithm used for the analysis.
- configuration: details about the parameters of the algorithm.
- report: scheme containing a representation of each resource analyzed.

Future works will include a more detailed parameterisation of the algorithms performed during the analysis process.

2.4.2 List of Supported Annotations

Topic A topic is an abstract concept described by a sorted list of words that represents a research area or subject in a domain. The order means the relevance of the word in the topic. It contains the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `topics/de305d54-75b4-431b-adb2-eb6b9e546014.` URIs from external services describing synsets (e.g. BabelNet) will also be used in future.
- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- content: an account of the meaning of the topic. Usually, it is the top 15 relevant words.

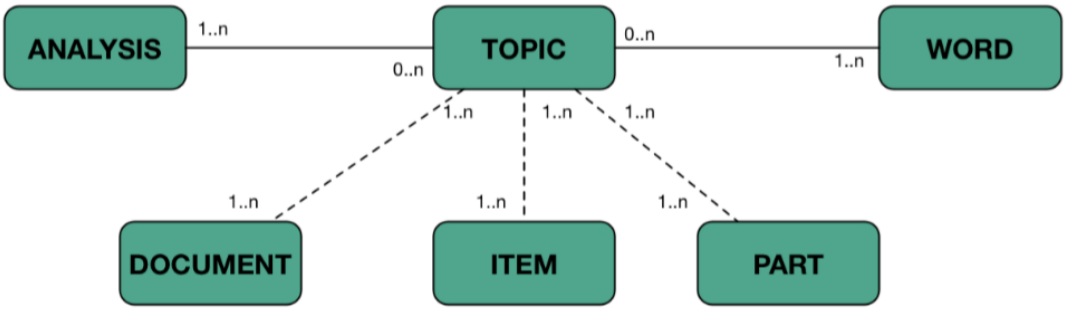


Figure 2: Topics and their relations with the different Resources

Furthermore, a topic contains one or more words and one or more analyses. In turn, a word can be referenced by zero more topics and an analysis can be referenced by zero or more topics. Since topics are also used to represent resources, they are also referenced by one or more document, item and parts.

Term A Term is an concept that is relevant or pertinent for describing the knowledge underlying the data in the corpus. Terms are generated by the learning module, and they include the following information:

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g `terms/de305d54-75b4-431b-adb2-eb6b9e546014.` URIs from external services describing synsets (e.g. BabelNet) will be also used in future.
- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF8601.

Relation A relation is an associative or semantic link between two resources. In the current version of DRInventor Platform, they are mainly used for three purposes: 1) to establish relations between words (normally entities, those relations are called associative), 2) to indicate properties between terms (semantic relations, including antonymy, meronymy, etc.) and 3) to formalise similitudes between resources based on topics they contain. The latter will be used for generating recommendations.

- uri: the Uniform Resource Identifier created by the system to uniquely identify it. It must be a Universally Unique Identifer (UUID) along with a prefix identifying the type of the resource: e.g relations/de305d54-75b4-431b-adb2-eb6b9e546014
- creation-time: date on which this resource was created. It must be a formatted timestamp following ISOF8601.
- type: associative, semantic, similarity, etc.
- describes: more details about the relationship, e.g. antonymy, meronymy, etc.

Furthermore, a relation contains two words and one or more analyses. In turn, a word can be referenced by zero or more relations and an analysis can be referenced by zero or more relations.

2.5 Exploring Resources and Annotations in the DRInventor Platform

In order to explain how the different resources and annotations inside DRInventor can be accessed after the ingestion of RO's in external repositories have finished, we will showcase how to browse the information corresponding to the SIGGRAPH (short for Special Interest Group on Computer GRAPHics and Interactive Techniques) corpus, a set of approximately 1500 research papers that has been used to test the features of our framework in different experiments and publications. SIGGRAPH is one of the top annual conferences on computer graphics (CG) convened by the ACM SIGGRAPH organization and it is attended by tens of thousands of computer professionals. The indexed set includes the papers presented at this conference between 2002 and 2016.

In the following subsections we exemplify how to glance and browse those SIGGRAPH resources and annotations (Documents, Parts, Terms, Topics...) via two different methods: a GUI based application named the DRInventor Repository dashboard, and the REST API that allows to programmatically retrieve the information from the platform by different agents.

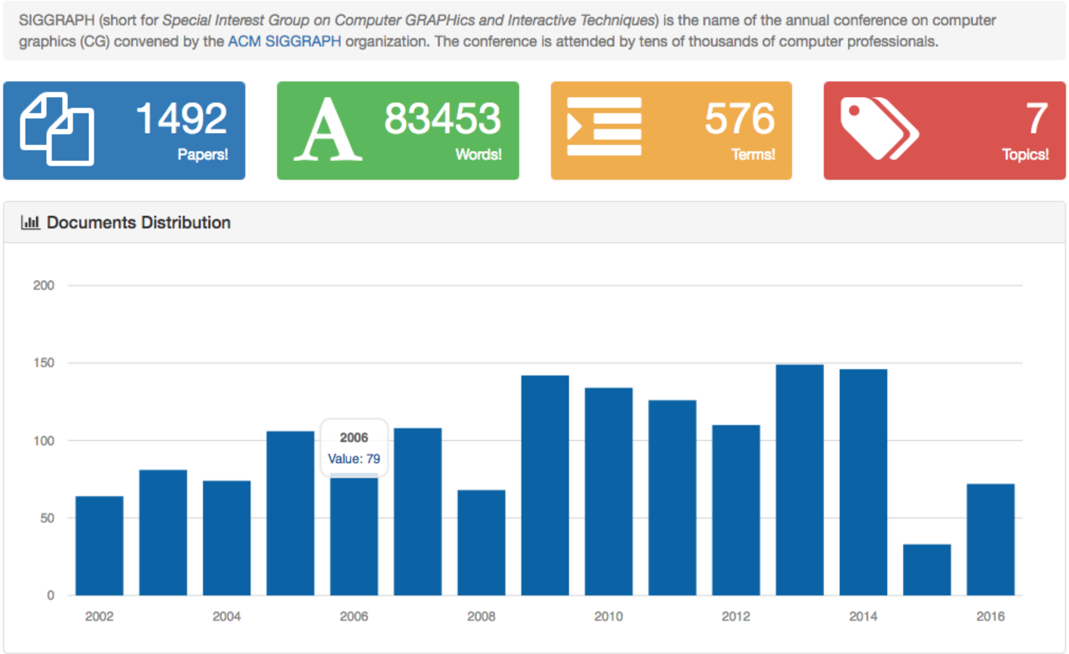
2.5.1 The SIGGRAPH Corpus via the DRInventor Dashboard

Users and experts aiming to explore the resources available in the different corpuses inside DRInventor can use the dashboard to obtain a quick overview of the resources indexed inside the framework. The prototype of this dashboard is depicted in Figure 3 and it is available at `http://drinventor.dia.fi.upm.es/`

Figure 3: Landing page of DRInventor Dashboard

By clicking on the option *Explore* → *Corpus* we can access a summary of the resources and annotations inside the selected corpus. Figure 4 shows how the current snapshot of the SIGGRAPH corpus contains 1492 documents, 83453 Words, 576 Terms, and 7 different topics detected. Under the panel called ‘Document Distribution” we can glance how the number of documents published in the conference has varied over the years, from 2012 to 2016. In case we need to further details on a particular document, we can click on them to see information such as the publication date, or the break down of parts inside the scientific paper (See

Figure 5).



SIGGRAPH (short for *Special Interest Group on Computer GRAPHics and Interactive Techniques*) is the name of the annual conference on computer graphics (CG) convened by the ACM SIGGRAPH organization. The conference is attended by tens of thousands of computer professionals.

1492 Papers!

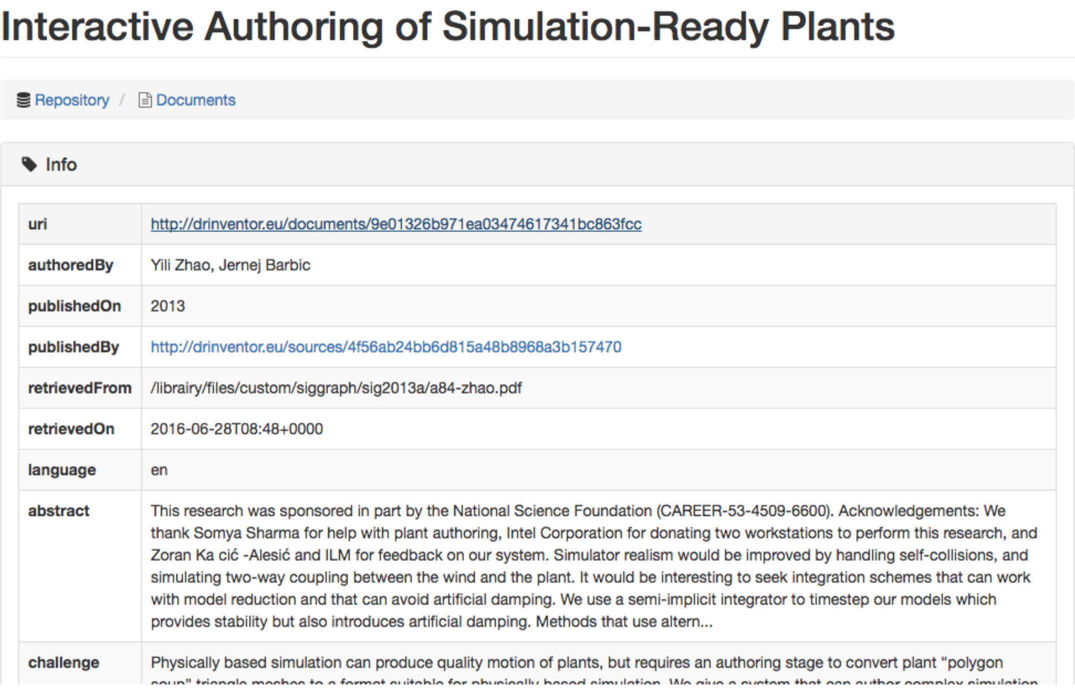
83453 Words!

576 Terms!

7 Topics!

Documents Distribution

Figure 4: Dashboard summarising SIGGRAPH corpus statistics on documents and annotations



Interactive Authoring of Simulation-Ready Plants

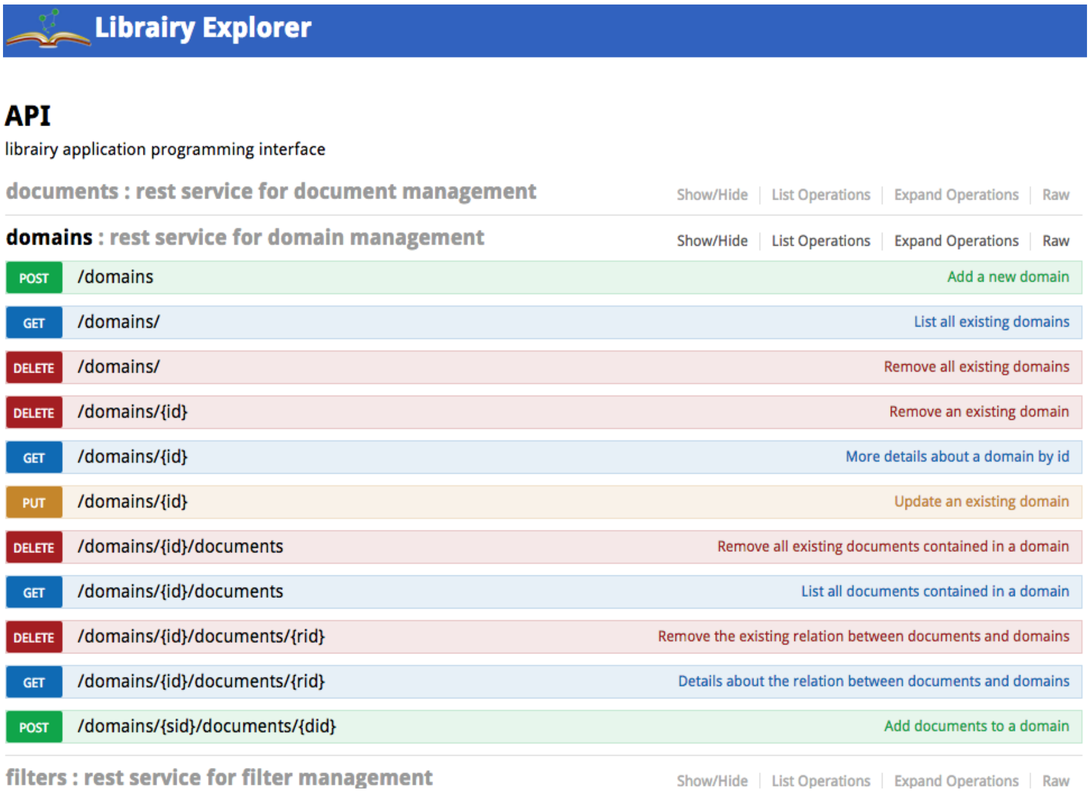
uri	http://drinventor.eu/documents/9e01326b971ea03474617341bc863fcc
authoredBy	Yili Zhao, Jernej Barbic
publishedOn	2013
publishedBy	http://drinventor.eu/sources/4f56ab24bb6d815a48b8968a3b157470
retrievedFrom	/librairy/files/custom/siggraph/sig2013a/a84-zhao.pdf
retrievedOn	2016-06-28T08:48+0000
language	en
abstract	This research was sponsored in part by the National Science Foundation (CAREER-53-4509-6600). Acknowledgements: We thank Somya Sharma for help with plant authoring, Intel Corporation for donating two workstations to perform this research, and Zoran Ka cić -Alesić and ILM for feedback on our system. Simulator realism would be improved by handling self-collisions, and simulating two-way coupling between the wind and the plant. It would be interesting to seek integration schemes that can work with model reduction and that can avoid artificial damping. We use a semi-implicit integrator to timestep our models which provides stability but also introduces artificial damping. Methods that use altern...
challenge	Physically based simulation can produce quality motion of plants, but requires an authoring stage to convert plant "polygon soup" triangle meshes to a format suitable for physically based simulation. We give a system that can author complex simulation

Figure 5: Details about a particular research paper inside the Platform

2.5.2 The SIGGRAPH Corpus via the REST API

The DRInventor dashboard provides a human-friendly way of accessing the data available in the platform, therefore helping the users to browse the information in a more intuitive way. However the same data can be programatically accessed via a REST API[20] that provides different methods in order to retrieve/upload the different information stored in by DRInventor.

In this section we present the most relevant API calls exposed by the platform in order to obtain similar information than the one offered in the dashboard. This way different Web agents can access not only the data but also different corpus / document processing techniques that can contribute to the implementation of innovative tools for visualising the data or exploiting the results of the DRInventor features in new and innovative ways. The API implemented for DRInventor is based on the Swagger toolbox[21], so the different methods can be also invoke through the corresponding Swagger UI instance as depicted in Figure 6.



Librairy Explorer

API

librairy application programming interface

documents : rest service for document management

domains : rest service for domain management

POST	/domains	Add a new domain
GET	/domains/	List all existing domains
DELETE	/domains/	Remove all existing domains
DELETE	/domains/{id}	Remove an existing domain
GET	/domains/{id}	More details about a domain by id
PUT	/domains/{id}	Update an existing domain
DELETE	/domains/{id}/documents	Remove all existing documents contained in a domain
GET	/domains/{id}/documents	List all documents contained in a domain
DELETE	/domains/{id}/documents/{rid}	Remove the existing relation between documents and domains
GET	/domains/{id}/documents/{rid}	Details about the relation between documents and domains
POST	/domains/{sid}/documents/{did}	Add documents to a domain

filters : rest service for filter management

Figure 6: The SwaggerUI exposing the DRInventor REST API

Accessing Resources Resources like Documents, Parts, or Items can be accessed via the corresponding API calls. There are methods for listing the entire set of elements being addressed, for getting the details of a particular instance, for generating new individuals and for deleting existing ones. Below we include the main methods to manage Documents, the

[20] http://drinventor.dia.fi.upm.es/api/

[21] `http://swagger.io/`

calls corresponding to other type of resources can be easily derived from them.

→ Listing the documents available in the corpus:

```
Url: http://drinventor.dia.fi.upm.es/api/0.2/documents/
Parameters: []
Method: GET
Response:
[
  "http://drinventor.eu/documents/39d9c4c0bb38b5fd740be63ad4cbb82c",
  "http://drinventor.eu/documents/fe95c24777e690d9ea8aedce1fe8610e",
  "http://drinventor.eu/documents/470c8134092ab394ee4590089add40bf",
  "http://drinventor.eu/documents/2c743b3f8d576a0145909d2f6fdca138",
  "http://drinventor.eu/documents/4638670749c720d87e6f95d3e4b91729",
  "http://drinventor.eu/documents/65401490542663e3b902ece90710e455",
  "http://drinventor.eu/documents/cda7c9724e8f8f1a9d8134c274a72900",
  ...
]
```

→ Getting the details about a particular document inside the corpus (e.g. with UUID fe95c24777e690d9ea8aedce1fe8610e):

```
Url: http://drinventor.dia.fi.upm.es/api/0.2/documents/fe95c24777e690d9ea8aedce1fe8610e
Parameters: []
Method: GET
Response:
{
  "uri": "http://drinventor.eu/documents/fe95c24777e690d9ea8aedce1fe8610e",
  "creationTime": "2016-06-28T08:40+0000",
  "publishedOn": "2014",
  "publishedBy": "http://drinventor.eu/sources/4f56ab24bb6d815a48b8968a3b157470",
  "authoredOn": "2014",
  "authoredBy": "Yun Teng, Miguel A. Otaduy, Theodore Kim",
  "retrievedFrom": "/librairy/files/custom/siggraph/sig2014a/a106-teng.pdf",
  "retrievedOn": "2016-06-28T08:40+0000",
  "language": "en",
  "title": "Simulating Articulated Subspace Self-Contact",
  "description": "Any opinions, findings, and conclusions...",
  "type": "research-paper"
}
```

→ Adding a new document inside the corpus:

```
Url: http://drinventor.dia.fi.upm.es/api/0.2/documents/
Parameters:
{
  "uri": "http://drinventor.eu/documents/f895f3223c04005784d1a2f236a5a637",
  "creationTime": "2016-06-28T08:41+0000",
  "publishedOn": "2014",
  "publishedBy": "http://drinventor.eu/sources/4f56ab24bb6d815a48b8968a3b157470",
  "authoredOn": "2014",
  "authoredBy": "Vladimir G. Kim, Siddhartha Chaudhuri, Leonidas J. Guibas, Thomas A.
      Funkhouser",
  "retrievedFrom": "/librairy/files/custom/siggraph/sig2014a/a120-kim.pdf",
  "retrievedOn": "2016-06-28T08:41+0000",
  "language": "en",
  "title": "Shape2Pose: Human-Centric Shape Analysis",
  "description": "This project was supported by NSF grants DMS...",
  "type": "research-paper"
  "format": "PDF"
}
Method: POST
Response:
{
  "uri": "http://drinventor.eu/documents/f895f3223c04005784d1a2f236a5a637",
  "creationTime": "2016-06-28T08:41+0000",
  "publishedOn": "2014",
  "publishedBy": "http://drinventor.eu/sources/4f56ab24bb6d815a48b8968a3b157470",
  "authoredOn": "2014",
  "authoredBy": "Vladimir G. Kim, Siddhartha Chaudhuri, Leonidas J. Guibas, Thomas A.
      Funkhouser",
  "retrievedFrom": "/librairy/files/custom/siggraph/sig2014a/a120-kim.pdf",
  "retrievedOn": "2016-06-28T08:41+0000",
  "language": "en",
  "title": "Shape2Pose: Human-Centric Shape Analysis",
  "description": "This project was supported by NSF grants DMS 1228304...",
  "type": "research-paper"
}
```

→ Deleting a document inside the corpus (e.g. with UUID d41d8cd98f00b204e9800998ecf8427e):

```
Url: http://drinventor.dia.fi.upm.es/api/0.2/documents/d41d8cd98f00b204e9800998ecf8427e
Parameters: []
Method: DELETE
Response: []
```

Accessing Annotations Once the Resources indexed in the platform after ingesting the Research Objects from the external repositories, different Analysis processes are launched in order to generate annotations that further describe them and support advanced operation over the data, such as Terms or Topics. Those annotations are also available through the REST API, following a similar REST schema than the one designed for the Documents, therefore being able to access the entire list of annotations, retrieve one in particular, create new instances or delete a previously existing ones. But apart from those basic manipulation operations, annotations have different methods to manage how they are attached to the different resources they describe. Below we show some examples of how it is possible to check on those connections or create new ones, for the particular case of Topics.

→ Check the list of Words a Topic is composed of (taking as example the topic with UUID 6982c58129104ed592ef00365b0c408f):

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/topics/6982c58129104ed592ef00365b0c408f/words
Parameters: []
Method: GET
Response:
[
  "http://drinventor.eu/words/point",
  "http://drinventor.eu/words/surface",
  "http://drinventor.eu/words/figure",
  "http://drinventor.eu/words/method",
  "http://drinventor.eu/words/image",
  "http://drinventor.eu/words/algorithm",
  "http://drinventor.eu/words/sample",
  "http://drinventor.eu/words/time",
  "http://drinventor.eu/words/function",
  "http://drinventor.eu/words/graphics"
]
```

→ Check the details about the relationship between a particular Word and the Topic it belongs to (taking as example the topic with UUID 6982c58129104ed592ef00365b0c408f and the word “surface”):

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/topics/6982c58129104ed592ef00365b0c408f/words
    /surface
Parameters: []
Method: GET
Response:
{
  "uri": "http://drinventor.eu/mentions/6982c58129104ed592ef00365b0c408f-surface",
  "creationTime": "2016-06-28T09:15+0000",
  "weight": 0.011329446747521116
}
```

→ Obtain the list of Topics associated to a particular document (taking as example the document with UUID 415ad00f7ea3ac00108c09a2ae1a2b95):

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/documents/415ad00f7ea3ac00108c09a2ae1a2b95/
    topics
Parameters: []
Method: GET
Response:
[
  "http://drinventor.eu/topics/acb0972c27d6096c4d6b6f89c15a44af",
  "http://drinventor.eu/topics/8617654279396256e3f97933205e9807",
  "http://drinventor.eu/topics/d6fd22f2f4735923b89cde588972bacf",
  "http://drinventor.eu/topics/6982c58129104ed592ef00365b0c408f",
  "http://drinventor.eu/topics/29c200fae24ffdf1f03c78854a012f2e",
  "http://drinventor.eu/topics/e541b4b026d8542f4c0c995f1911db04",
  "http://drinventor.eu/topics/28ea36b7c2d287cf9a70aa5aaf3a55d"
]
```

Accessing Platform Features Resources and Annotations are the main data structures represented inside the repository of DRInventor. However, the REST API is also able to provide not only access modification for those instances, but also execute the different exploration and browsing techniques that will be introduced in next Section 3 so they are available to third parties interested in leveraging on those advanced operations without having to reimplement them or duplicating the data into an additional Librairy instance.

→ Given a pair of documents D1 and D2, find a list of intermediate documents that define a path between them. Taking as example the documents UUID’s 39d9c4c0bb38b5fd740be63ad4cbb82c and 470c8134092ab394ee4590089add40bf, this operation is divided into two different REST

API calls included below. The first one is intended to specify the beginning and ending documents and therefore defining the opposite sides of the path, and the second one retrieves the list of intermediate documents that allows to consecutively jump from the initial resource to the final one, based on the similarity between them. For more details about how this similarity is calculated, see Section 3.3:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/paths
Parameters:
{
  "end": "39d9c4c0bb38b5fd740be63ad4cbb82c",
  "start": "470c8134092ab394ee4590089add40bf"
}
Method: POST
Response:
{
  "uri": "http://drinventor.eu/paths/415211ecb6ac38efbf941f35412cca30",
  "creationTime": "2016-08-18T12:47+0000",
  "start": "470c8134092ab394ee4590089add40bf",
  "end": "39d9c4c0bb38b5fd740be63ad4cbb82c"
}
```

The second REST API call takes as input the UUID of the generated path (415211ecb6ac38efbf941f35412c for invoking the intermediate document finding operation, providing a list of the selected candidates as showing in the response field:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/paths/415211ecb6ac38efbf941f35412cca30/
    documents
Parameters:
Method: GET
Response:
[
  {
    "weight": 0,
    "resource": "http://drinventor.eu/documents/c3ed94476a9c83b574c2daa3ee1ce5e1",
    "description": "Energy Redistribution Path Tracing"
  },
  {
    "weight": 0.9526461184343431,
    "resource": "http://drinventor.eu/documents/634b9381c3325ffa0229490308b4a097",
    "description": "Unifying Points, Beams, and Paths in Volumetric Light Transport
        Simulation"
  },
  {
    "weight": 0.7420375067669466,
    "resource": "http://drinventor.eu/documents/645fb91fd74aae3d3e5d781f9a1ef607",
    "description": "Joint Importance Sampling of Low-Order Volumetric Scattering"
  },
  {
    "weight": 0.8583685840008418,
    "resource": "http://drinventor.eu/documents/b129a61be216cca71a35044522bd72d1",
    "description": "Unbiased, Adaptive Stochastic Sampling for Rendering Inhomogeneous
        Participating Media"
  },
  {
    "weight": 0.7399280266839483,
    "resource": "http://drinventor.eu/documents/5ad4fffc2f9a0c97bc3d7ba06a801f27",
    "description": "Real-time Soft Shadows in Dynamic Scenes using Spherical Harmonic
        Exponentiation"
  },
  {
    "weight": 0.7541165865268739,
    "resource": "http://drinventor.eu/documents/748c1515f899bf891e3c09d793a1ee4c",
    "description": "Interactive Hair Rendering Under Environment Lighting"
  },
  {
    "weight": 0.7373365936139138,
    "resource": "http://drinventor.eu/documents/bdf9e7ba8beb111c93dd7f1d507dc6bb",
    "description": "Simulating and compensating changes in appearance between day and night
        vision"
  }
]
```

3 Probabilistic Topic Modelling for Generating RO's Relations

Having the different research resources indexed into the DRInventor platform as shown in the previous section 2, we need to provide different mechanisms for agents accessing the framework in order to retrieve relevant pieces of information in a timely manner. In order to achieve this goal, we have implemented a set of context and content-based [20] recommendations of research objects that allow scientists and experts in the domain to navigate through the knowledge in an effective manner, discovering new pertinent facts that can bring value to their research without having to deal with the complexity of manually browsing huge collections of documents.

In the first subsection 3.1 we include an introduction to probabilistic topic modelling, a technique that allows to programatically generate different topics which are relevant for a certain collection of textual documents, and assign to each of those documents the combination of those topics that better fits its content. In the next subsection 3.2 we will show how to optimise the configuration of our topic modelling approach based on LDA [7]. Finally in subsection 3.3 we explain how we can leverage on different annotations and specially the previously generated topics, in order to determine the similarity between resources and therefore generate connections between research objects materialised as instances of the class Relation (see section 2.4.2).

3.1 Introduction to Probabilistic Topic Modelling

In order to provide content-based recommendation, the first thing we need to define is how the indexed RO's will be featured. Traditional retrieval tasks over large collection of textual documents [15] highly rely on individual features like term frequencies (TF-IDF). However, new ways of characterising documents based on the automatic generation of models highlighting the main subjects covered in the corpus have been appearing during the last years. *Probabilistic Topic Modelling* [6] algorithms are statistical methods that analyse the words of the original texts to discover the themes that run through them, how those themes are connected to each other, and how they change over the time. Originally developed as a text-mining tool, topic models are now being used to detect instructive structures in data such as genetic information, images and networks, and they also have applications in other fields such as bioinformatics.

One of the main advantages is that they do not require any prior annotations or labelling of the documents. The topics emerge, as hidden structures, from the analysis of the original texts. The "topics" produced by topic modelling techniques are clusters of similar words. A topic model captures this intuition in a mathematical framework, which allows examining a set of documents and discovering, based on the statistics of the words in each, what the topics might be and what each document's balance of topics is.

Those topics offer a much more intuitive, yet sophisticated way of performing knowledge discovery tasks in big collections of documents and therefore have served as the basis to implement recommendation tasks in DRInventor Platform. We have leveraged the inferred hidden structure of the collection for analysing each document in the corpus and producing annotations that ease RO's classification, and corpus exploration. Compared to traditional text based searches, topic models can organise the collection according to the discovered themes that are more intuitive to browse, and better correspond to what a human intrinsically expects when navigating the knowledge (See Figure 7).

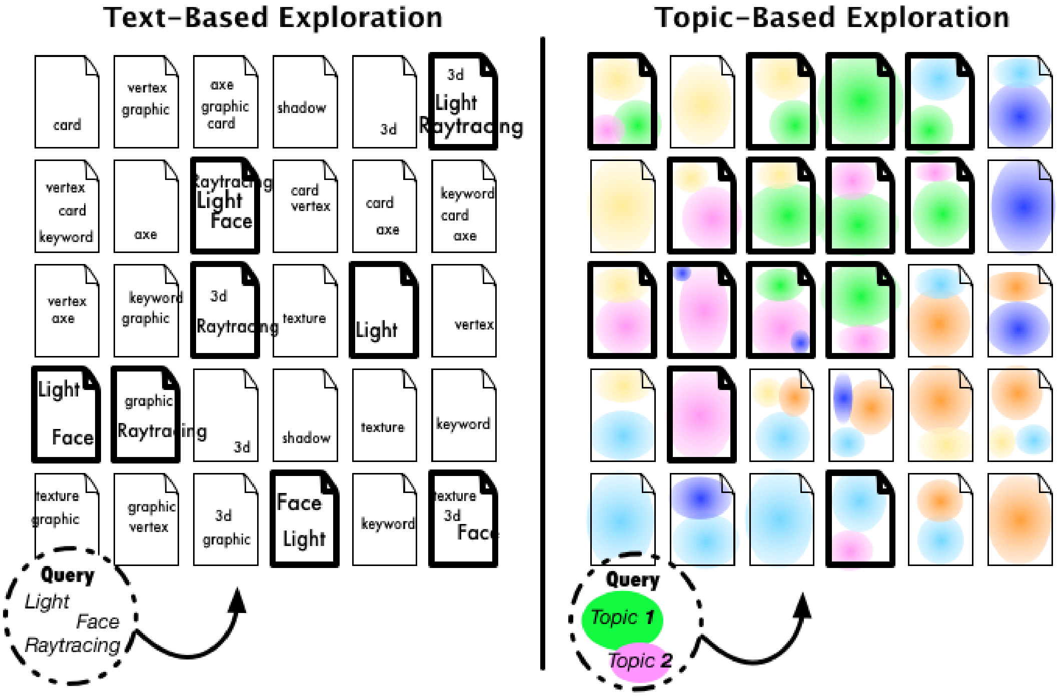


Figure 7: Textual based exploration tasks directly rely on the occurrences and frequency of words in the documents. Topics provide a more human friendly navigation that organises the documents into subjects that promote corpus awareness and knowledge discovery

Even though the generated topics have an enormous potential to support those kind of retrieval operations, it is still difficult how to further interpret the generated hidden structure that the topic are describing, and to determine how these annotations can be used to deeper identify relationships between resources. In this section we will elaborate on how topic modelling provides us an algorithmic solution to organising large collections of research objects and provide useful recommendation operations.

3.1.1 Latent Dirichlet Allocation

The most commonly used generative topic model is *latent Dirichlet allocation* (LDA) [7]. This and other topic models such as *Probabilistic Latent Semantic Analysis (PLSA)* [16] are part of the larger field of *probabilistic modelling*. They are well-known latent variable models for high dimensional count data, such as text data in the *bag-of-words* representation or any other count-based data representation but, while LDA has roots in LSA and PLSA (it was proposed as a generalisation of PLSA), it was cast within the generative Bayesian framework to avoid some of the overfitting issues that were observed with PLSA. As mentioned before, since PLSA is a *discriminative model*, it is unable to describe topics, i.e. hidden structures, but LDA is able to build a generative model to avoid that limitation.

In generative probabilistic modelling, data is treated as arising from a generative process that includes *hidden variables*. This generative process defines a *joint probability distribution* over both the observed (O) and hidden random variables (μ). Then data is analysed by using that joint distribution to compute the *conditional distribution* of the hidden variables given the observed variables $p(\mu \mid O)$. This conditional distribution is also called the *posterior distribution*. In LDA, the observed variables are the words of the documents, the hidden variables are the topic structure and the generative process is the problem of computing the posterior distribution, i.e. the conditional distribution of the hidden variables given the documents:

$$p(O,\mu) = p(O \mid \mu) \cdot p(\mu) = p(\mu \mid O) \cdot p(O) \tag{1}$$

This statistical model tries to capture the intuition that documents exhibit multiple topics. Each document exhibits the topic in different proportion, each word in each document is drawn from one of the topics, where the selected topic is chosen from the per-document distribution over topics. All the documents in the collection share the same set of topics, but each document exhibits these topics in different proportion. Documents are each represented as a vector of counts with W components, where W is the number of words in the vocabulary. Each document in the corpus is modelled as a mixture over K topics, and each topic k is a distribution over the vocabulary of W words. Each topic is drawn from a Dirichlet with parameter β, while each document is sampled from a Dirichlet with parameter α. Formally, a *topic* is a multinomial distribution over words of a fixed vocabulary representing some concept.

The Dirichlet distribution is a continuous multivariate probability distribution parameterized by a vector of positive reals whose elements sum to 1. It is *continuous* because the relative likelihood for a random variable to take on a given value is described by a probability density function, and also it is *multivariate* because it has a list of variables whose values are unknown. In fact, the Dirichlet distribution is the conjugate prior of the categorical distribution and multinomial distribution.

From a collection of documents, LDA infers: *per-word* topic assignment, *per-document* topic proportions and *per-corpus* topic distributions. Exact inference, i.e. computing the posterior over the hidden variables, for this model is intractable [7], then a variety of approximate algorithms have been proposed [3] such as *collapsed Gibbs sampling (CGS)*, *variational Bayesian inference (VB)*, *collapse variational Bayesian inference (CVB)*, *maximum likelihood* *estimation (ML)* and *maximum a posteriori (MAP)*.

Unlike a clustering model, where each document is assigned to one cluster, LDA allows documents to exhibit multiple topics (see right side of Figure 7). For example, LDA can capture that one article might be about “biology” and “statistic”, while another might be about "biology" and "physics". Since LDA is unsupervised, the themes of “physics”, “biology” and “statistics” can be discovered automatically from the corpus; the mixed-membership assumptions lead to sharper estimates of word cooccurrence patterns.

3.1.2 Topic Visualisation in DRInventor

Later in Section 4.4 we will introduce how DRInventor Dashboard offers some graph-based visualisations of documents where certain parts of the plots had been coloured according to the predominant topic they are associated to (see Figure 23). This way, the GUI can provide a visual representation of how the documents are spread into different thematics that together build up the SIGGRAPH corpus.

The topics in the platform have been obtained by launching an implementation of the LDA algorithm configured with optimal parameters k, α and β as will be explained in Section 3.2. For the particular case of this corpus, the number of topics suggested by the optimisation algorithm has been 7. However, we can obtain more insights about how those topics are composed by accessing a dedicated tab under *Explore* → *Topics*.

In Figure 8 we can see a bar chart representing the volume of documents per topic generated for the current corpus. This visualisation gives the user an idea about the topics which are highly present inside the set of research objects ingested and therefore are very representative of the corpus, and those which are predominant in just a very few documents (such as Topics 3 or 7) because they correspond to more specific areas or emerging fields of research.

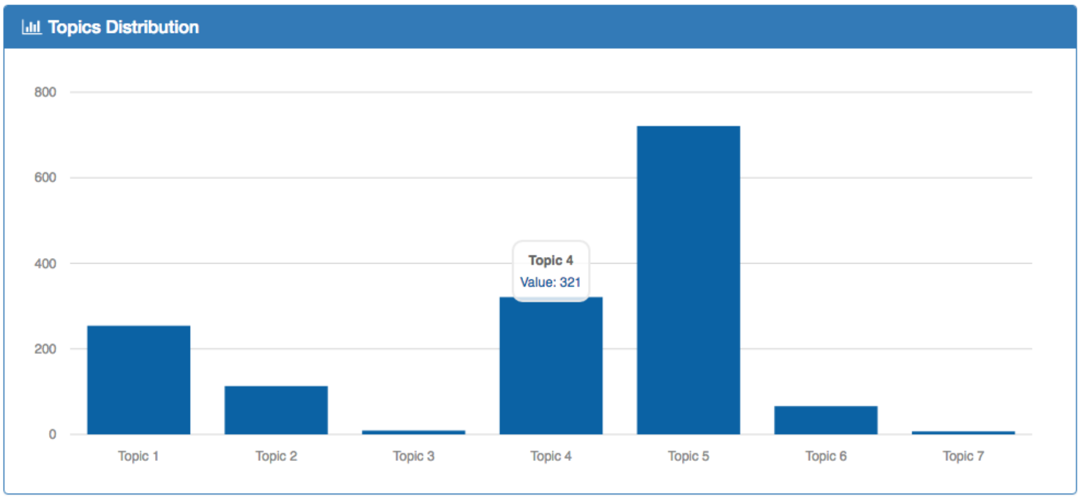
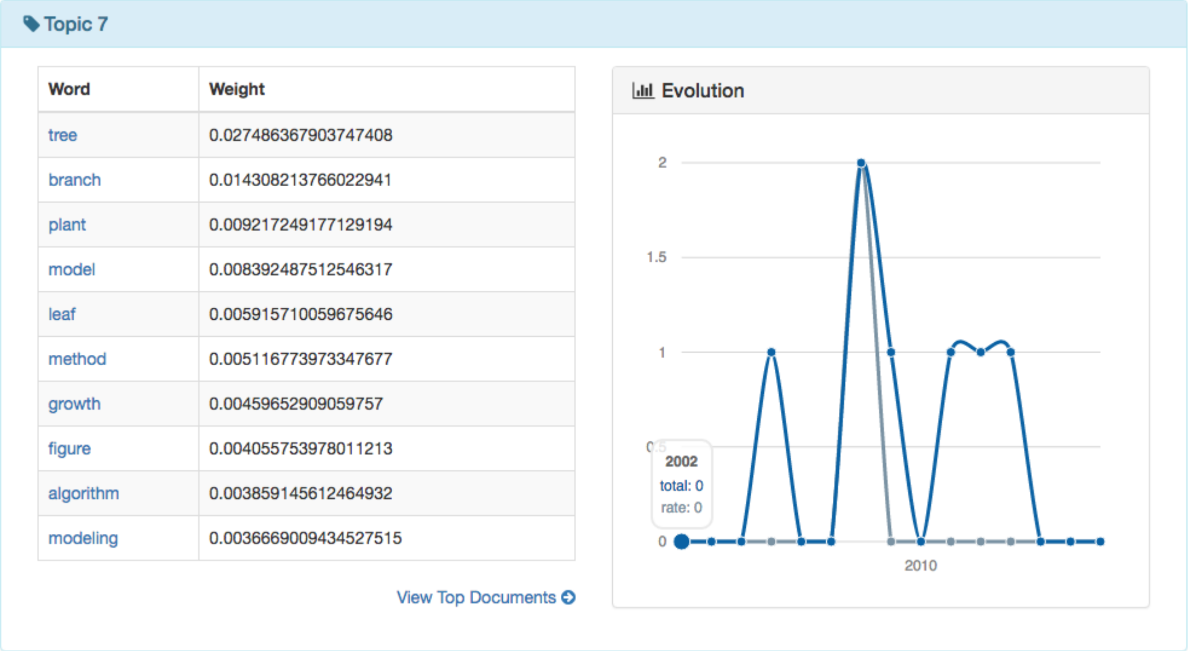


Figure 8: Distribution of Documents per SIGGRAPH corpus

In addition we can look deeper into each individual topics, in order to find out what subject they can refer to, or how they have evolved over time. On the left side of Figure 9 we have access to the list of higher ranked words according to the LDA process having generated the topic 7. Each of those words are accompanied by a numerical score giving a more precise idea about the relevance of the word inside the topic. For the current example, the words “Tree”,“Branch” and “Plant” are placed in top positions, indicating that this topic has a lot to do with the graphic representation of vegetation and forestry elements. By clicking over the option “View Top Documents” we obtain a list of the documents where this topic is more prominent. The titles, e.g. “Capturing and Animating the Morphogenesis of Polygonal Tree Models”, we can confirm that the top papers clustered under this topic are strongly related with the aforementioned subjects.



Topic 7

Word	Weight
tree	0.027486367903747408
branch	0.014308213766022941
plant	0.009217249177129194
model	0.008392487512546317
leaf	0.005915710059675646
method	0.005116773973347677
growth	0.00459652909059757
figure	0.004055753978011213
algorithm	0.003859145612464932
modeling	0.0036669009434527515

Figure 9: Details about Topic 7: main words describing it and temporal evolution of the topic through the last years (2002 - 2016)

Finally, on the right side of Figure 9 we can check the temporal evolution of the topic through the last years (2002 - 2016). The blue coloured line indicates absolute amount of documents under that particular topic, by year. The grey line represents the normalisation of the former number by the total amount of documents published that year in the conference, giving a better insight about the relative importance of this topic during a particular period of time, compared to the volume of papers in others subareas of computer graphic. The peaks in the plot represent periods of time where this topic has grown in importance. For example, we can see how during year 2008 there is a higher rate of research works about this subject than in previous and subsequent periods.

Finally, it is also possible to see the distribution of topics for a particular document D by relying on the visualisations that the DRInventor Platform offers about particular instances of RO's. For example if we access the information about the research paper *Interactive Authoring of Simulation-Ready Plants*, we can see how a circular chart with one axe per topic quickly allows us to know which are the most prominent topics inside the paper (See Figure 10). At the same time in a tile-based diagram, we can also check how the most relevant words per each topic are distributed inside the paper being considered (see Figure 11)

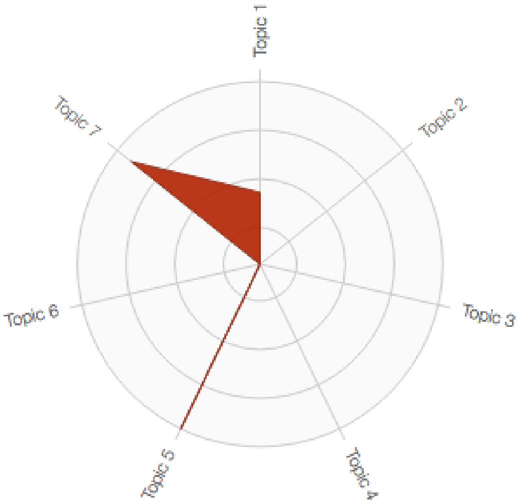


Figure 10: Most predominant topics in document *Interactive Authoring of Simulation-Ready Plants*

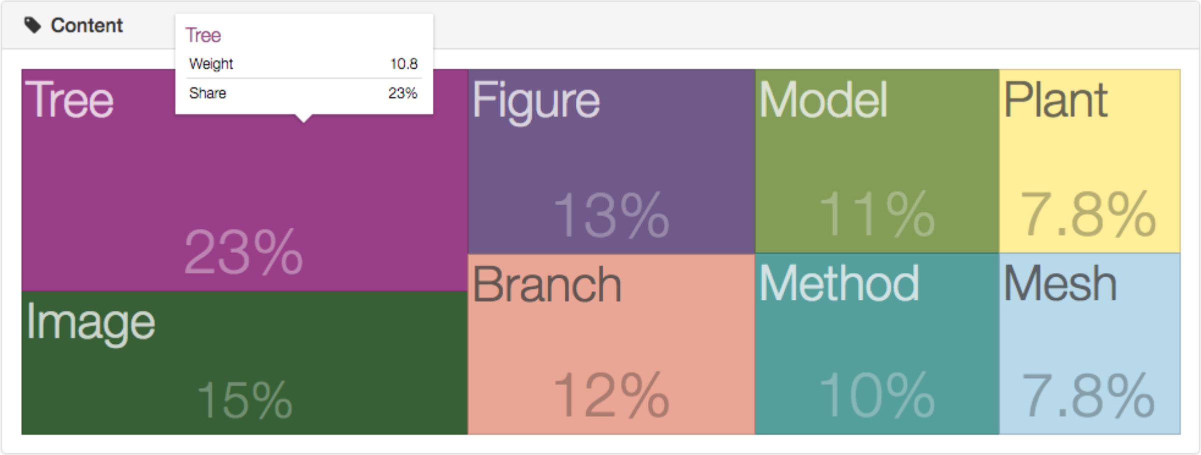


Figure 11: Relative Dominance of Topical Words inside document *Interactive Authoring of Simulation-Ready Plants*

3.2 Configuring LDA using an Evolutionary Multi-objective Optimisation

As mentioned in Section 3.1.1, a LDA model is used to describe the inherent topic distribution of existing textual resources, in our case *research objects*. This model requires some parameters to be selected, and they need to be properly adjusted to obtain higher quality models.

Since LDA is characterised by Dirichlet distributions of topics and documents, i.e. multivariate generalisation of the Beta distribution, it is parameterised by two positive shape parameters, α and β, that appear as exponents of the random variable and control the shape of the distribution. Moreover, the dimensionality of each Dirichlet distribution needs to be fixed. So the dimensionality value of the Dirichlet distribution of topics is known and equals to the size of the vocabulary. However, the dimensionality of the Dirichlet distribution of documents, i.e. number of topics, is not necessarily known beforehand and needs fixed.

Thus, we need to estimate three parameters: the *number of topics (k)*, the concentration parameter (α) for the prior placed on documents' distributions over topics, and the concentration parameter (β) for the prior placed on topics and distributions over terms. Some authors [3] have proposed inferences to calculate these parameters, however the implementation of LDA made by Spark (based on *Expectation/Maximization*) and used by RESSIST does not admit these values yet.

In addition, all parameters are corpus-level parameters, so we need to calculate new values whenever the corpus changes. From the point of view of efficiency, this operation is executed in background mode each time a group of resources are added. The size of that group is defined beforehand.

3.2.1 Multi-Objective Evolutionary Approach

New values of *log-likelihood* and *log-prior* are obtained for each new LDA execution that measures the goodness of the model. The higher these values are, the better the model fits. For this reason, having several conflicting objetives, i.e. improvement of one objective may lead to deterioration of another, and having parameters to estimate (k, α and β), a *Multi-Objetive Evolutionary Algorithm (MOEA)* is used to find the *Pareto* optimal solution.

A single solution optimizing *log-likelihood* and *log-prior* simultaneously does not exist. Instead, the best trade-off solution called *Pareto optimal* will be obtained. Taking into account performance behaviour [27] and to prevent new objectives derived from the use of the model, the *Non-Sorting Genetic Algorithm-III (NSGA-III)* [12] is chosen and the optimisation problem is defined as follows:

- **Objectives**:
  - ○ $\min \| \log(likelihood) \|$
  - ○ $\min \| \log(prior) \|$
- **Constraints**:

- $5.1 < \alpha < 20.0$: Document Concentration. It represents the distribution of a document in topics. That is, how specific is a document. The lower boundary of α (5.1) is greater than the higher boundary of β (5.0) because, in our opinion, if a term belongs to more than one topic, a document will contain equal or more number of topics than those contained in the term. Moreover, we have considered that the number of topics in a document is, at least, 4 times greater than the topics contained in a term, for that reason the higher boundary is 20.0 for α and 5.0 for β.
- $1.0 < \beta < 5.0$: Topic Concentration. It represents the distribution of a topic over terms. That is, how a term can belong to several topics. In our opinion, a term can only belong to no more than 5 topics, because greater values will create more ambiguous models.
- $0 < k < 2 * \sqrt{p/2}$: Number of topics. Usually around the root square of the half of population (p).

- **Crossover**: Motivated by the success of binary-coded genetic algorithms in problems with discrete search space, the operator selected was *Simulated Binary Crossover (SBX)* [10] that solves problems having a continuous search space instead of binary. This operator has a search power similar to that of the single-point crossover. It was set to 0.9 to facilitate the explorative capacity of the algorithm.

- **Mutation**: Mutation operators have been utilized extensively in MOEAs as solution variation mechanisms. Mutation operators assist to the better exploration of the search space [18]. Different approaches have been proposed depending on the representation used in MOEAs such as binary or real values. In this case, the operator selected was the *Polynomial Mutation* operator [11] [13] which allows big jumps in the search space of the decision variable, escaping from local optima and modifying a solution when on the boundary. It was set to 1.0 to promote the explorative analysis.

- **Selection**: The global best solution is selected by a *N-ary Tournament* operator. This operator prefers feasible solutions over infeasible solutions (for constraint handling), non-dominated solutions over dominated solutions (for handling multiple objetives) and less-crowed solutions over more-crowded solutions (for the maintenance of diversity).

The boundaries of the constraints have been defined, as previously mentioned, according to the implementation of the LDA algorithm made by Spark. The minimum value of the parameters, *alpha* and *beta*, is defined to 1.0 by default, but they will be different in our application. Since the *beta* parameter describes the concentration of topics in words, we consider only low values (lower than 5.0 and greater than 1.0) trying to get more representative words for each topic. Defining this range of values, the algorithm will avoid using the same words to characterize different topics, getting distinguished distributions of topics in documents. Moreover, defining high values for the *alpha* parameter (between 5.1 and 20.0), the algorithm considers that a document can contain more than one topic, but these distributions will not be smooth. We are looking for a characterisation that enables us to handle more than one topic in a document, but also enough differences between the topic distributions of different documents to group documents that are talking about the same area or in the same way.

The number of topics will be between 1 and an empirical value defined by the root square of the half of population (p). This approximation is useful to avoid a high exploration during the learning process focusing on a smaller set of values.

3.2.2 Experiment: Optimising LDA for a Corpus of Research Objects

Trying to measure the learning process, we executed the evolutionary algorithm implemented by the JMetal framework [17] on a corpus previously created by the *Hoarder* and the *Harvester* applications (for more information please refer to [4]). The corpus has been created using the *Hoarder*[22] tool, configured to retrieve research objects from some OAI-PMH data providers published by Innovare Journal[23]. The final set of resources is composed by 100 research objects balanced over 10 different research areas: Agricultural Science (IJAGS), Business Management (IJBM), Education (IJOE), Ayurvedic Science (IJAS), Engineering and Technology (IJET), Health Science (IJHS), Life Science (IJLS), Medical Science (IJMS), Social Science (IJSS) and Science (IJS).

We used the implementation of LDA developed by Apache Spark [2] that learns the model using *Expectation-Maximization (EM)* on the likelihood function ($p(O \mid \mu)$). The values of *logLikelihood* and *logPrior* are obtained from that model using the research objects included in the corpus.

A first analysis consisted on executing the learning process setting a maximum number of executions to 30 and a maximum number of LDA iterations to 20. The rest of values (topics, alpha and beta) were dynamically obtained by the NSGA-III algorithm trying to optimize the final values of *LogLikelihood* and *LogPrior*. The results are shown in Table 1:

Taking into account the constraints of the parameters, the number of *topics* for this corpus is limited between [1-14] (for a population of 100 individuals, the $2 * \sqrt{p/2}$ is equals to 14) , the *alpha* value between [5.1-20.0] and the *beta* value between [1.1-5.1]. Then, according to the results, the value of *beta* is the most stable, only varying twice while the value of *topics* is the most scattered. This behaviour shows that only taking into account the values of *LogLikelihood* and *LogPrior*, a LDA model using 11 topics may have a similar accuracy to another that uses only 6. It occurs because the concentration of topics in a document (*alpha* value) and the concentration of topics in a word (*beta* value) are different in both cases. So, for the LDA model that uses 11 topics, the value of *alpha* is 12.1 and the value of *beta* 1.1, while for the model that uses 6 topics, the value of *alpha* is equal to 6.1 and the value of *beta* is 1.1. Reasoning

[22] `https://github.com/cbadenes/epnoi-harvester`

[23] `http://innovareacademics.in/`

Test	Topics	Alpha	Beta	LogLikelihood	LogPrior	MaxIters	Time(ms)
1	4	9.6	1.1	$-392.121,13$	$-4,40$	$30/20$	217.849
2	10	7.9	1.1	$-381.553,12$	$-10,70$	$30/20$	277.912
3	6	5.1	1.1	$-386.208,24$	$-6,60$	$30/20$	275.010
4	6	6.1	4.1	$-418.703,32$	$-178,14$	$30/20$	167.890
5	3	6.1	1.1	$-395.317,26$	$-3,36$	$30/20$	310.983
6	11	12.1	1.1	$-385.879,43$	$-11,71$	$30/20$	188.025
7	1	13.1	2.1	$-411.222,50$	$-11,32$	$30/20$	275.841
8	3	7.1	1.1	$-395.034,58$	$-3,37$	$30/20$	364.952
9	8	14.4	1.1	$-385.927,65$	$-8,66$	$30/20$	290.385
10	2	7.1	1.1	$-400.680,65$	$-2,24$	$30/20$	215.063

Table 1: LDA configurations suggested by the NSGA-III algorithm after 30 evaluations of 20 executions

Test	Topics	Alpha	Beta	LogLikelihood	LogPrior	MaxIters	Time(ms)
1	2	6.9	1.1	$-401.676,96$	$-2,24$	$500/20$	671.791
2	2	6.1	1.3	$-402.490,28$	$-6,41$	$500/20$	619.366
3	12	5.4	1.1	$-378.598,41$	$-12,7$	$500/20$	781.803
4	9	6.7	1.1	$-380.974,83$	$-9,71$	$500/20$	870.969
5	7	6.2	1.1	$-387.861,83$	$-7,56$	$500/20$	499.501

Table 2: LDA configurations suggested by the NSGA-III algorithm after 500 evaluations of 20 executions

about this, when the number of topics is low, the concentration of topics in documents is also low, because the topics in that case are more general than when there are more of them. In these cases they are more specific.

As expected, the best configuration defines 10 topics, i.e the same number of different research areas included in the corpus, with a level of concentration of topics in documents (*alpha*) equals to 7.9 and a level of concentration of words in topics (*beta*) equals to 1.1.

All these tests are executed with a maximum number of iterations for NSGA-III equal to 30 and a maximum number of iterations for LDA equal to 20. Trying to discover whether the first value, i.e. max iterations for NSGA-III, can affect to the final result we have executed the same algorithm increasing it to 500. We considered that value because an evolutionary algorithm needs many executions to explore the population. The results are showed in the Table 2.

Now, the best configuration (highest *loglikelihood*=$-380.974,83$) defines 9 topics, 6.7 of *alpha* concentration and 1.1 of *beta* concentration. However, the parameters show the same behaviour as before, being the topics and *alpha* values the most scattered values. No im-

Test	Topics	Alpha	Beta	LogLikelihood	LogPrior	MaxIters	Time(ms)
1	11	5.3	1.1	$-377.300,52$	$-11,74$	$200/100$	1.276.498
2	7	10.9	1.3	$-394.008,71$	$-21,36$	$200/100$	1.342.360
3	7	5.1	1.8	$-401.114,06$	$-54,60$	$200/100$	1.204.178
4	12	6.1	1.1	$-376.704,06$	$-12,72$	$200/100$	1.455.811
5	12	5.9	1.1	$-377.411,04$	$-12,76$	$200/100$	1.167.862

Table 3: LDA configurations suggested by the NSGA-III algorithm after 200 evaluations of 100 executions

provement was detected increasing only the maximum number of iterations of NSGA-III, so we decided to increase also the number of iterations of LDA to 100 and reducing the maximum for NSGA-III to 200. Then, we obtained more accurate results, as shown in Table 3.

These results show that the evolutionary algorithm as well as the LDA Model require a high number of iterations. Now, the values of *LogLikelihood* are usually better than before, and the best configuration appears in the case: 12 topics, *alpha* equals to 5.9 and *beta* equals to 1.1 to create a model with an accurate equals to $-377.411,04$. We used this configuration for the rest of evaluations.

Note the variability of execution time. This is because we introduced a small cache in the NSGA-III algorithm to avoid executing LDA configurations that had been previously executed.

3.3 Topic-Based Resource Similarity

As stated previously, the system will make predictions, inferences and recommendations based on research objects and any other useful information derived from them. For this some metrics are required, mainly similarity measures, to connect resources, authors and extract knowledge from these relationships. Those connections found between resources are materialised into the repository in the form of instances of the class *Relation* as specified in Section 2.4.2.

Measuring the similarity of ROs is a key task from which to obtain useful knowledge. Its definition must be general enough to overcome the particular characteristics of the different types of resources that ROs aggregate. Thus, similarity evaluations may show differences on the accuracy between *regular-resources*, *conceptual-resources* or *topical-resources* but not between different types of content in the same resource expression, i.e. a textual-based *regular-resource* and an image-based *regular-resource*.

Within the scope of the DRInventor framework and the research objects indexed in the platform, we have identified two kind of information describing them:

- **context-based**, i.e. authors, license rights, formats, etc annotating the RO.
- **content-based**, i.e. text, image, code.. shaping up the RO.

The similarity measure that we propose in this deliverable (sim_D) leverages on both aspects of the RO's available in the platform by considering a weighted sum of *context-based* similarity (sim_{ctx}) and *content-based* similarity (sim_{cont}):

$$sim_D(R_i,R_j) = \alpha * sim_{cont}(R_i,R_j) + (1-\alpha) * sim_{ctx}(R_i,R_j) \tag{2}$$

where $\alpha \in [0,1]$

Depending on the nature of the resource, i.e. *regular-resource*, *conceptual-resource* or *topical-resource*, each of these *content-based* and *context-based* similarity measures are different as detailed above.

3.3.1 Content-based Similarity

Frequency Dimension

Both *Words-Space* and *Concepts-Space* are based on frequency vectors as feature vectors. The first space counts word frequencies and the second one counts concept frequencies. The expression that describes the content similarity between resources based on frequency vector is the same in both spaces.

A Resource in the platform, hereinafter called *regular-resource*, is an entity that may aggregate other/s *regular-resource/s* and contains identification and descriptive information such as title, authors and a bag-of-words describing its content. Regardless of whether it is a textual resource or an image resource or any other, it is annotated by a list of words to describe what the content means. Future works will take into account the type of the content to make a more specific similarity metric, but at the present time the *content-based* similarity measure considers that group of words to measure how similar two resources are.

Because a *regular-resource* may contain aggregated resources, the feature vector used to measure the similarity is the vectorial sum of the feature vectors of each nested resource. We used the *cosine similarity* based on the *Euclidean dot product* as similarity measure to take into account the proportional use of words instead of frequency values directly. So, the ***content-based similarity measure*** is:

$$sim_{cont}(R_i, R_j) = cos(\hat{r}_i, \hat{r}_j) \tag{3}$$

where $\hat{r}_i$ is the feature vector of the research object R_i described as *regular-resource* or *conceptual-resource* and $cos(\hat{r}_i, \hat{r}_j)$ is the *cosine similarity*:

$$cos(P,Q) = cos(\theta) = \frac{P \cdot Q}{\|P\| \|Q\|} = \frac{\sum\limits_{i=1}^{n} P_i \times Q_i}{\sqrt{\sum\limits_{i=1}^{n} (P_i)^2} \times \sqrt{\sum\limits_{i=1}^{n} (Q_i)^2}} \tag{4}$$

Topic Distribution Dimension

However, in the *Topics-Space* the feature vector is a topics distribution expressed as vector of probabilities. When a resource is aggregated by other resources, the *bag-of-concepts* used to calculate the topics distributions is the sum of concepts used in each nested resource. For this reason, the topics distribution of the root of an aggregation is enough to describe the complete research object.

Taking into account this premise, the similarity measure between two *topical-resources* will be based on the distance between their topics distributions. Since they are Dirichlet distributions (probability mass functions), the measure used was the *Jensen-Shannon divergence*, which can be defined as the average of the *Kullback-Leibler (KL) divergence* between them. KL has two major problems: in the case that one of topics distribution is zero, KL is not defined and it is not symmetric, what does not fit well with semantic similarity measures which in general are symmetric [23]. To solve these problems, *Jensen-Shannon divergence* considers the average of the distributions as below [8]:

$$JSD(p,q) = \sum_{i=1}^{T} p_i * \log \frac{2 * p_i}{p_i + q_i} + \sum_{i=1}^{T} q_i * \log \frac{2 * q_i}{q_i + p_i} \tag{5}$$

where T is the number of topics and p, q are the topics distributions

Our ***content-based similarity measure*** use the *Jensen-Shannon divergence* transformed into a similarity measure as follows [9]:

$$sim_{cont}(R_i, R_j) = 10^{-JSD(p,q)} \tag{6}$$

where R_i, R_j are the research objects and p, q the topics distributions of each of the *topical-resources* describing them.

3.3.2 Context-based Similarity

Currently, context-based similarity is only related to author-based similarity. The plan is to include more elements in the future, both extracted directly from the resource or inferred, so as to increase the accuracy of the measure. But now the system must work properly with authors:

$$sim_{ctx}(R_i, R_j) = sim_{authors}(R_i, R_j) \tag{7}$$

Before obtaining the similarity of authors we need to define how an author is described. Similar to feature vectors to describe the content of a resource, an author is represented by a vector that describes adequately his/her most relevant aspects to allow us to take measures between them. The dimension of this vector will depend on the cardinality of the used space. As in the case of resources, two types of feature vectors exist: frequency-based and topics-based.

Frequency Dimension

This similarity will be used both in *Words-Space* and *Concepts-Space* because it considers that an author is described by each of the feature vectors of the research objects published by him/her. Recently, a temporal combination has been proposed to obtain a valid similarity measure between authors [19] . They defined an *author similarity (AS)* based on *cosine similarity* of the feature vectors for a given interval time:

$$AS_{cos}(A, B, t_1, t_2) = cos(\sum_{i=t_1}^{t_2} \hat{a}_i, \sum_{i=t_1}^{t_2} \hat{b}_i) \tag{8}$$

where $\hat{a}_i$ and $\hat{b}_i$ are the feature vectors of the authors A and B in the i-th year.

Authors usually publish more than one research object in the same year, so the feature vector for that i-th year will be the vectorial sum of feature vectors of each research object published.

However, this metric does not take into account possible common shifts of interests of the authors. In fact, if the author A worked on topic T_1 and then shifted to topic T_2, he will be considered similar to author B who was originally in T_2 and then moved to T_1. To avoid this problem, a metric that pays attention to the period of time in which an author addresses a specific topic is needed, rewarding common trajectories. Hence, in order to strengthen the importance of the time factor, a partial similarity recursively on increasingly shorter time intervals is proposed and the final similarity is the average of the results. More formally, a *temporal author similarity*

(TAS) between an author A and an author B in the interval $t_1 - t_2$ is:

$$TAS(A,B,t_1,t_2) = \frac{\sum_{i=0}^{m}\left[(\sum_{j=0}^{2^i-1} AS(A,B,t_1 + \lceil \frac{j\cdot(t_2-t_1)}{2^i} \rceil, t_1 + \lfloor \frac{(j+1)\dot{(}t_2-t_1)}{2^i}))/2^i\right]}{m+1} \tag{9}$$

where $m = \lfloor log(t_2 - t_1) \rfloor$

This temporal author similarity covers well the case in which both authors are present in the same time interval, however an author may have no publications in some of the years inside the interval. Then a penalty P is applied as the average of AS of n authors randomly extracted from the input. In our opinion, this penalty should be changed by the feature vector of the last publication, then the intervals of time without publications would have the same feature vector than the last year with publication. Thus, our similarity measure between two authors is:

$$sim_{author}(A,B) = TAS(A,B,t1,t2) \tag{10}$$

where t_1 is the oldest publication date of both authors and t_2 the newest one.

Once we know the similarity measure between two authors, we can calculate the ***author-based similarity measure*** between ROs as the minimum similarity value between the authors of each research object:

$$sim_{authors}(R_i,R_j) = min(sim_{author}(a_{im},a_{jn})) = min(TAS(A,B,t1,t2)) \tag{11}$$

where a_{im} is the m-th author of the *regular | conceptual-resource* i, and a_{jn} the n-th author of the *regular | conceptual-resource* j.

The feature vectors used in this calculus contain frequencies of words in *Words-Space* and frequencies of concepts in *Concepts-Space*.

Topic Distribution Dimension

Now the feature vector is a multinomial probability distribution so the challenge here is to produce a consensus topics distribution for each author by combining appropriately the topics distributions of their publications. The most popular choice for this aggregation is *Linear Pooling*, which assigns each individual forecast a weight which reflects the importance of the publication, but if we provide an equal weight to every probability the method reduces to an arithmetic average. A *Generalized Linear Pooling* extends the previous approach considering the possibility of negative weights. However [21] any linear combination of (calibrated) forecast is uncalibrated and lacks sharpness then a *Beta-transformed Linear Pooling* is proposed applying a *Beta transformation* to linear pooling operators in order to add a recalibration step to the process and improve their performance. A probability $P_G(A)$ is said to be calibrated if $P(Y_k|P_G(A_k)) = P_G(A_k, k = 1 \ldots K)$ [21]. Sharpness refers to the concentration of the aggregated distribution. The more concentrated it is, the sharper it is.

Intuitively, aggregation operators based on multiplication seem more appropriate than those based on addition. *Log-linear Pooling* is a linear operator of the logarithms of the probabilities that does not preserve independence and does not verify the marginalization property.*Generalized Logarithmic Pooling* extends it by adding an arbitrary bounded function. On the other hand, instead of establishing a pooling formula from an axiomatic point of view, the aggregation of two distributions could be those that share properties (moments or conditional probabilities) and minimize the KL divergence between them.

Furthermore, as showed in several simulation studies [1], *linear pooling* performs poorly relative to other pooling formulas with a multiplicative instead of an additive structure. Also, many of non-linear methods involve a large number of parameters, making them computationally complex and susceptible to over-fitting. By contrast, parameter-free approaches, such as the median or the geometric mean of the odds, are too simple to be able to incorporate the use of training data optimally.

Recently, an approach based on the *log-odds statistical model* of the data has been proposed [25] being an alternative way to express probabilities using the odds ratio.

However, the LDA model considers topics distributions as Dirichlet distribution, i.e continuous multivariate probability distributions , then we can combine them to get a more general topics distribution using the Bayes' Theorem. Thus, considering the following topics distributions (td_1, td_2) for the research objects (R_1, R_2) and the topics (T_1, T_2, T_3):

$$td_1 = (t_{11}, t_{12}, t_{13})$$
$$td_2 = (t_{21}, t_{22}, t_{23})$$

and taking into account that:

$$t_{ij} = p(T_i/R_j)$$

the consensus topics distribution td_f will be:

$$td_f = (P(T_1/R_1, R_2), P(T_2/R_1, R_2), P(T_3/R_1, R_2))$$

As R_1 and R_2 are independent and using the Bayes' theorem we get:

$$P(T_i/R_1, R_2) = \frac{P(R_1) \cdot P(R_2)}{P(R_1, R_2)} \times \frac{P(T_i/R_1) \cdot P(T_i/R_2)}{P(T_i)} = \alpha \times \frac{P(T_i/R_1) \cdot P(T_i/R_2)}{P(T_i)} \tag{12}$$

where α is a class-independent term depending only on the data. As we have measured these data, its value is not interesting here (we are not doing model comparisons), so we treat it as a normalization constant which ensures the Dirichlet constraint that $\sum_k P(T_k/R_1, R_2) = 1$.

Now that we know how to combine topic distributions, we can redefine the *author similarity (AS)* expression using the *Jensen-Shannon Divergence* as a distance measure of topics distributions and taking its similarity expression for a given interval time:

$$AS_{JSD}(A, B, t_1, t_2) = 10^{-JSD(\hat{a_{12}}, \hat{b_{12}})} \tag{13}$$

where $\hat{a_{12}}$ and $\hat{b_{12}}$ are the consensus topics distributions of authors A and B for the interval of time $t_1 - t_2$.

Topic	Most Frequent Terms
0	’collect’ ,’extract’ ,’procedur’ ,’evalu’ ,’univers’ ,’plant’ ,’chemic’ ,’health’ ,’medicin’ ,’found’ ,’standard’ ,’research’ ,’antimicrobi’ ,’revis’ ,’receiv’ ,’pharmaci’ ,’abstract’ ,’keyword’ ,’accept’ ,’screen’
1	’found’ ,’review’ ,’email’ ,’scienc’ ,’articl’ ,’accord’ ,’receiv’ ,’keyword’ ,’abstract’ ,’revis’ ,’accept’ ,’import’ ,’which’ ,’other’ ,’introduct’ ,’refer’ ,’gmail’ ,’studi’ ,’innovar’ ,’journal’
2	’afford’ ,’itself’ ,’arrang’ ,’lesson’ ,’mobil’ ,’integr’ ,’citizen’ ,’reach’ ,’strong’ ,’charg’ ,’allow’ ,’equip’ ,’altern’ ,’opportun’ ,’start’ ,’provis’ ,’build’ ,’offer’ ,’challeng’ ,’subject’
3	’defin’ ,’first’ ,’review’ ,’receiv’ ,’abstract’ ,’keyword’ ,’articl’ ,’revis’ ,’accept’ ,’other’ ,’point’ ,’which’ ,’refer’ ,’group’ ,’journal’ ,’innovar’ ,’paper’ ,’inform’ ,’anoth’ ,’conclus’
4	’email’ ,’where’ ,’articl’ ,’revis’ ,’keyword’ ,’abstract’ ,’accept’ ,’which’ ,’introduct’ ,’refer’ ,’innovar’ ,’journal’ ,’engin’ ,’through’ ,’gener’ ,’variou’ ,’conclus’ ,’receiv’ ,’consid’ ,’techniqu’
5	’method’ ,’should’ ,’receiv’ ,’abstract’ ,’keyword’ ,’revis’ ,’accept’ ,’research’ ,’about’ ,’effect’ ,’introduct’ ,’which’ ,’studi’ ,’refer’ ,’perform’ ,’achiev’ ,’journal’ ,’innovar’ ,’aspect’ ,’materi’
6	’develop’ ,’signific’ ,’other’ ,’effect’ ,’should’ ,’through’ ,’introduct’ ,’refer’ ,’which’ ,’receiv’ ,’abstract’ ,’keyword’ ,’increas’ ,’accept’ ,’requir’ ,’innovar’ ,’journal’ ,’articl’ ,’revis’ ,’total’
7	’agent’ ,’treatment’ ,’therefor’ ,’further’ ,’method’ ,’result’ ,’present’ ,’system’ ,’scienc’ ,’articl’ ,’differ’ ,’revis’ ,’receiv’ ,’keyword’ ,’abstract’ ,’respect’ ,’accept’ ,’effect’ ,’activ’ ,’absorb’
8	’colleg’ ,’patient’ ,’treatment’ ,’email’ ,’result’ ,’articl’ ,’research’ ,’receiv’ ,’abstract’ ,’keyword’ ,’revis’ ,’accept’ ,’object’ ,’indor’ ,’qualiti’ ,’ayurveda’ ,’group’ ,’manag’ ,’increas’ ,’method’
9	’gmail’ ,’email’ ,’receiv’ ,’revis’ ,’accept’ ,’studi’ ,’journal’ ,’innovar’ ,’district’ ,’rural’ ,’articl’ ,’abstract’ ,’keyword’ ,’research’ ,’primari’ ,’patient’ ,’medic’ ,’occur’ ,’intern’ ,’adult’
10	’appli’ ,’respect’ ,’nation’ ,’evalu’ ,’second’ ,’countri’ ,’properti’ ,’intern’ ,’univers’ ,’those’ ,’sampl’ ,’resist’ ,’express’ ,’howev’ ,’includ’ ,’process’ ,’function’ ,’number’ ,’basic’ ,’avail’
11	’discuss’ ,’method’ ,’result’ ,’receiv’ ,’abstract’ ,’keyword’ ,’accept’ ,’found’ ,’introduct’ ,’studi’ ,’refer’ ,’scienc’ ,’journal’ ,’innovar’ ,’articl’ ,’email’ ,’differ’ ,’revis’ ,’under’ ,’india’

Table 4: Distribution of terms by topics

3.3.3 Similarity Measure sim_D over a RO Corpus: Experiments and Results

At this point, we have a *Topics-Space* composed by *TopicalResources*, and created from the *ConceptualResources* that were directly generated from the *RegularResources* that describe the research objects of the corpus. In this space, an LDA model is generated to create the *TopicalResources* that contain *ConceptualResources* along with topics distributions. Using these distributions, the RESSIST application can calculate the similarity measures based on topics between a couple of resources, i.e. between two research objects.

Applying the configuration of the LDA model suggested by the learning algorithm detailed in Section3.2, i.e. 12 topics, alpha=6.1 and beta=1.1, our application builds a Topic Model that defines probabilistic distributions for each resource based on its *bag-of-concepts* (in fact, as before mentioned, based on words). After a stemming process, the concepts are reduced to their stem, i.e. base or root. Taking the list of stems for each resource and their frequencies, the system built 12 topics that contain, in a different proportion, the list of stems of the corpus. This distribution of *stems* by topics is listed in the table 4, showing only the 20 most relevant stem for each topic.

Using these topics, i.e probabilistic distributions of stems (from concepts/words), the model assigns a distribution of topics for each resource, Table 6, then our recommender system creates a similarity matrix calculating the similarity measurements between all the research objects of the corpus. The Table 5, for example, shows a column of that matrix. It contains the similarity measurements between the referenced research object, with a similarity equal to 1, and the rest of research objects of the corpus. In the table we also identify the data provider where the resource was published to identify the research area where the publisher, in this

Resource	Similarity	Research Object	Provider
1	1.0	SIMULTANEOUS ESTIMATION OF IRBESARTAN AND ATORVASTATIN BY FIRST ORDER DERIVATIVE SPECTROSCOPIC METHOD IN THEIR SYNTHETIC MIXTURE USE IN HYPERTENSION CONDITION	IJS
2	0.9985002911336149	DEVELOPMENT AND VALIDATION OF ANALYTICAL METHOD FOR IRBESARTAN AND ATORVASTATIN BY SIMULTANEOUS EQUTION SPECTROSCOPIC METHOD	IJS
3	0.9984274995642390	SIMULTANEOUS ESTIMATION OF IRBESARTAN AND ATORVASTATIN BY Q ABSORPTION RATIO METHOD IN THEIR SYNTHETIC MIXTURE USE IN CARDIAC CONDITION	IJS
4	0.9950251177695804	ARTEMETHER LUMEFANTRINE LOADED LIPOSPHERES EVALUATION OF PROPERTIES OF SOLUTOL HS 15 AND SOLUPLUS ON THE IN VITRO PROPERTIES	IJS
5	0.9890510173748973	EFFECT OF LYOPHILIZATION ON THE PHYSICOCHEMICAL AND PHYSICOTECHNICAL PROPERTIES OF ASPIRIN-LOADED LIPOSPHERES	IJS
6	0.9812091925351933	COMPATIBILITY OF BEAUVERIA BASSIANA (BALS.) VUILL ISOLATES WITH SELECTED INSECTICIDES AND FUNGICIDES AT AGRICULTURE SPRAY TANK DOSE	IJAGS
7	0.9487017007511238	DIFFERENT MODELS TO EVALUATE ANTIMICROBIAL AGENTS-A REVIEW	IJLS
8	0.9365837211037017	PREPARATION OF CHITOSAN STABILIZED OFLOXACIN- GOLD NANO CONJUGATE FOR THE IMPROVED ANTI BACTERIAL ACTIVITY AGAINST HUMAN PATHOGENIC BACTERIA	IJMS
9	0.8089238194923263	An overall review on Obesity and its related disorders	IJLS
10	0.21158453507688826	RESEARCH ON FORMULATION AND EVALUATION OF INSITU MUCOADHESIVE NASAL GELS OF METOCLOPRAMIDE HYDROCHLORIDE	IJMS
..	..	..	...

Table 5: Similarity measures between research objects from the same data provider.

case *Innovare Journal*, has classified the resource: Agricultural Science (IJAGS), Business Management (IJBM), Education (IJOE), Ayurvedic Science (IJAS), Engineering and Technology (IJET), Health Science (IJHS), Life Science (IJLS), Medical Science (IJMS), Social Science (IJSS) and Science (IJS). This classification is used as reference during the tests to check the validity of the results.

A first important behaviour showed in the table 5 is that, if considering as similar only the research objects with a similarity value greater than 0.5, only 8 research objects are similar to the referenced one, that is only the 8% of the corpus. This exhaustive classification, in our opinion, is caused by the high number of topics, 12, and the low value of *alpha*, 6.1. With these values the research objects have a low concentration of topics and then different research objects can be described by *strong* topic distributions, i.e high values for some topics and low for the rest, avoiding middle values. In fact, the difference between the 8th research object, *"An overall review on Obesity and its related disorders"*, and the 9th, *"RESEARCH ON FORMULATION AND EVALUATION OF INSITU MUCOADHESIVE NASAL GELS OF METOCLOPRAMIDE HYDROCHLORIDE"*, is almost 0.6 points. This behaviour is common for the rest of columns of the similarity matrix, as showed also in the table 7.

In addition, as expected, the most similar research objects are published in the same domain, IJS, so they belong to the same research area, Science. However, research objects from other research areas such as IJAGS (Agricultural Science), IJLS (Life Science) and IJMS(Medical Science) are also present in the column as similar. The reason is because our similarity measure does not handle the meaning of paragraphs or the relevance of terms considered by an author, it takes into account the frequency of terms to build a model based on their probabilities to belong to a topic, i.e. a cluster, and RESSIST uses these probabilities

R	S	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
1	ijs	0.0045	0.0049	0.0028	0.0044	0.0057	0.0044	0.0060	0.9459	0.0054	0.0034	0.0038	0.0081
2	ijs	0.0053	0.0056	0.0032	0.0051	0.0075	0.0052	0.0075	0.9342	0.0070	0.0038	0.0047	0.0105
3	ijs	0.0055	0.0060	0.0033	0.0051	0.0075	0.0049	0.0079	0.9338	0.0065	0.0039	0.0047	0.0103
4	ijs	0.0061	0.0061	0.0044	0.0060	0.0098	0.0053	0.0107	0.9259	0.0053	0.0041	0.0058	0.0098
5	ijs	0.0058	0.0070	0.0043	0.0071	0.0135	0.0059	0.0115	0.9138	0.0057	0.0045	0.0071	0.0131
6	ijags	0.0079	0.0095	0.0053	0.0092	0.0117	0.0079	0.0128	0.8981	0.0083	0.0064	0.0082	0.0141
7	ijls	0.0091	0.0132	0.0075	0.0139	0.0275	0.0097	0.0163	0.8621	0.0093	0.0091	0.0079	0.0140
8	ijms	0.0163	0.0113	0.0066	0.0081	0.0279	0.0102	0.0133	0.8495	0.0089	0.0082	0.0092	0.0299
9	ijls	0.0102	0.0238	0.0086	0.0211	0.0233	0.0334	0.0466	0.7382	0.0460	0.0170	0.0144	0.0168
10	ijms	0.0161	0.0192	0.0102	0.0149	0.0314	0.0156	0.0179	0.2156	0.0198	0.0123	0.0114	0.6151

Table 6: Distribution by topics of research objects listed in table 5

to make relationships between them based on their content and their contextual information. In our opinion this is useful, as showed later in Section 4.3, to discover relationships between papers touching different domains or research areas. For example in the table 5, the system detects that the research *"SIMULTANEOUS ESTIMATION OF IRBESARTAN AND ATORVASTATIN BY FIRST ORDER DERIVATIVE SPECTROSCOPIC METHOD IN THEIR SYNTHETIC MIXTURE USE IN HYPERTENSION CONDITION"* about Science is similar, with a similarity measure equals to 0.981209, to the research *"COMPATIBILITY OF BEAUVERIA BASSIANA (BALS.) VUILL ISOLATES WITH SELECTED INSECTICIDES AND FUNGICIDES AT AGRICULTURE SPRAY TANK DOSE"* about Agricultural Science because both publications contain, in a similar way, the set of terms described by the topics in the Table 4 , i.e. the words more frequently used are listed in topics 7, 11, 6 and 4, and the less frequently used are listed in topics 2, 9 and 10.

So, as showed in table 6, the topic 7 is the most representative for the research object *"SIMULTANEOUS ESTIMATION OF IRBESARTAN AND ATORVASTATIN BY FIRST ORDER DERIVATIVE SPECTROSCOPIC METHOD IN THEIR SYNTHETIC MIXTURE USE IN HYPERTENSION CONDITION "* , as well as the topics 11, 6 and 4. Then, research objects following a similar distribution of topics are more similar to it than the rest. This explains why a research object about Agricultural Science is more similar to a research object about Science than even other research objects also classified in Science. Depending on the stemming process, these similarities may vary, so that is a key task in our system. At this moment, we have used the Lucene classifier as the stemming algorithm, but in future work we will develop some variations to improve the accuracy of our classification procedure.

Moreover, as the Table 7 shows, the system is not influenced by the type of the data provider used to collect the research objects, i.e. by the research area where a research object is classified. The system has detected two research objects that are the same research object, *"CLINICAL-COMPARATIVE STUDY OF VIRECHAN & PAKSHAGHATARI GUGGULU ON PAKSHAGHAT W.R.S. TO HEMPIPLIGIA"*, but that they were published in two different data providers: IJLS and IJAS. This multiple classification express that we have considered in our model, a research object may be oriented to more than one topic.

It is important to mention that the research area where a research object is focused may

Similarity	Research Object	Provider
1.0	RIVIEW OF SHRINGA , ALABY AND CUPPOING THERAPY	IJAS
0.9901248961071275	PREVALENCE, ETIOLOGY AND CLINICAL FEATURES OF SKELETAL FLUOROSIS: A CRITICAL REVIEW.	IJMS
0.9898432754651388	ALL ABOUT YOGA	IJHS
0.9783233516492001	A CASE STUDY OF GIFTED CHILD.	IJOE
0.9752164237190362	ROLE OF AN IMPORTANCE OF ACTIVITIES IN SCHOOL ENVIRONMENT.	IJOE
0.9733055901744754	A STUDY OF EMOTIONAL INTELLIGENCE OF HIGHER SECONDARY SCHOOL TEACHERS OF MADHYA PRADESH	IJOE
0.9716959991552354	Relationship between cigarette smoking and body mass index in the Italian population	IJHS
0.9671450145579238	IMPACT OF ACTIVE LEARNING STRATEGIES TO ENHANCE STUDENT PERFORMANCE	IJOE
0.9474338200755947	DO LEADERSHIP QUALITIES DETERMINE COMPETENT PRINCIPALS	IJOE
0.9246011509355332	INFLUENCE OF ELECTRONIC MEDIA ON CHILDRENâĂŹS PERSONALITY DEVELOPMENT	IJSS
0.9154436939927828	AN INTRODUCTION OF PROBLEM BASED LEARNING IN IMS, BHU	IJMS
0.9064540067697475	An overall review on Obesity and its related disorders	IJLS
0.8396635494170802	KNOWLEDGE OF MEDICAL NEGLIGENCE AMONG MEDICAL STUDENTS	IJMS
0.27918402806682086	AYURVEDA AND MENTAL HEALTH	IJAS
..	..	...

Table 7: Similarity measures between research objects from different data providers.

be different from the content of the research object. The table 7 shows also a research object titled *"RIVIEW OF SHRINGA , ALABY AND CUPPOING THERAPY"* published in the *Innovare Journal of Ayurvedic Science* (IJAS) data provider that is more similar to other research object published in different data providers such as the *Innovare Journal of Medical Science* (IJMS) data provider, the *Innovare Journal of Health Science* (IJHS) or even the *Innovare Journal of Education* (IJOE). As previously mentioned, this occurs because our similarity measure is only based on the content (terms) and the authors of the research objects, instead of the research area or keywords. Future works will include a more complex semantic analysis to compose a similarity measure based not only in words or concepts, but also in the meaning of paragraphs or even in the ideas included in the conclusion section, for example.

The Figure 12 shows the graph created from the similarity matrix. Some clusters appear, as previously mentioned, because a high number of topics have been defined in the model.

3.3.4 Materialising Relations between Resources based on their Similarity

By relying on the Similarity Function between resources that we have just presented (sim_D) and establishing a threshold $T_r \in [0-1]$, we can decide to materialise the high degree of resemblance between pairs of documents in the form of instances of the class **Relations**, so that they are explicitly available in the repository for empowering other advanced operations that can leverage them.

For the particular case of the SIGGRAPH corpus in DRInventor, we did some empirical analysis by inspecting the similarity scores obtained when applying sim_D over a set of pre-selected pairs of documents we have read and therefore had a strong notion about the existence or not of such kind of connections. First results indicated that a $T_r = 0.5$ performed well, however given the sample of documents used for arriving at such a conclusion was small and probably not representative enough, we decided to materialise all the connections between

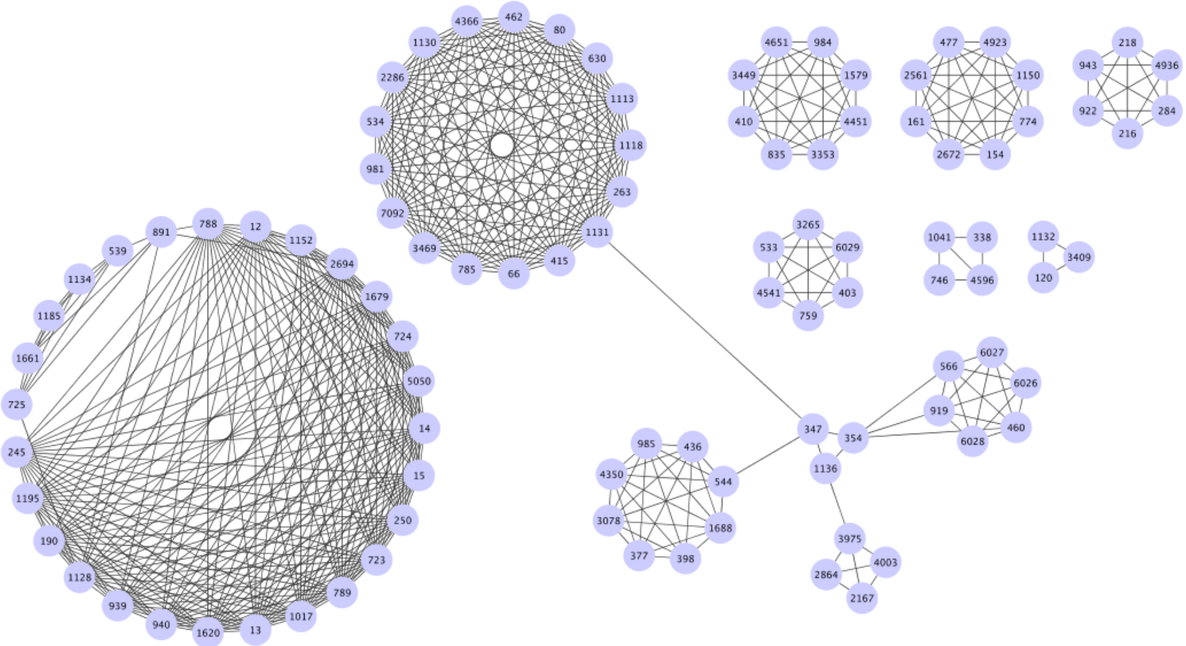


Figure 12: Research Object-Graph built from the test corpus

all possible pairs of documents in the corpus, and annotate them with the obtained similarity score, so that clients and third parties can decide on which threshold they use for filtering out irrelevant connections according to their objectives.

Besides, the platform has been configured to recalculate the relations between documents every given period of time (for the current configuration, 10 minutes) if a new Research object has been ingested so all the existing connections are updated accordingly.

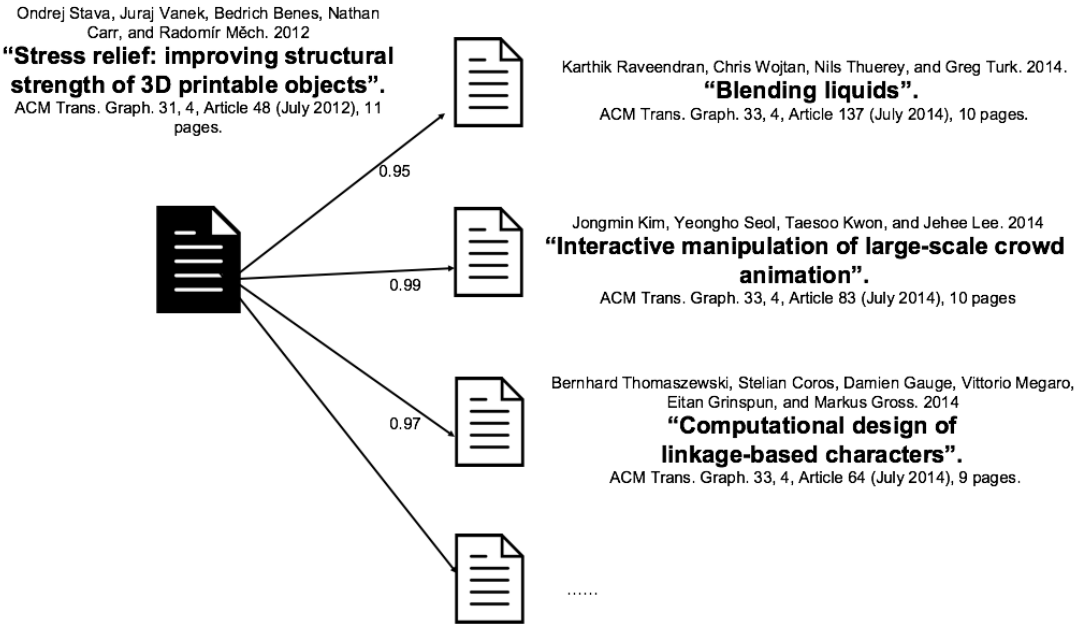
4 Recommendation Functionalities in DRInventor Platform

After applying the annotation techniques described in Section 3 over the corpora of SIGGRAPH resources indexed in DRInventor Platform, we are ready to exploit their results in order to offer more advanced features over the research objects, such as resource recommendations, or browsing of relevant scientific documents.

In this section we showcase different operations that are already available in the platform and intend to assist the research community in finding resources that can satisfy their different scientific needs. This set of operations leverages mainly the topics and in relations described in section 2.4.2 and previously generated between the documents at different levels of granularity. For each operation, we show how we can invoke the corresponding logic via both the REST API and the DRInventor Dashboard. In addition, in the last subsection 4.5 we advance some features that even if not explicitly exposed by the platform, may be feasible to implement by combining the existing logic and the annotations already available in the repository.

4.1 Get Relevant Resources given a Document

Given a particular document and thanks to the connections generated between documents in the DRInventor Platform, we can recommend a list of documents that are potentially interesting for the user given their similarity with the original one. Imagine, for example that a scientist has found the paper *"Stress relief: improving structural strength of 3D printable objects"* very interesting for his research and he would like to find documents addressing similar topics. This recommendation operation is illustrated in Figure 13.



Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomír Měch. 2012
"Stress relief: improving structural strength of 3D printable objects".
ACM Trans. Graph. 31, 4, Article 48 (July 2012), 11 pages.
0.95
0.99
0.97
Karthik Raveendran, Chris Wojtan, Nils Thuerey, and Greg Turk. 2014.
"Blending liquids".
ACM Trans. Graph. 33, 4, Article 137 (July 2014), 10 pages.
Jongmin Kim, Yeongho Seol, Taesoo Kwon, and Jehee Lee. 2014
"Interactive manipulation of large-scale crowd animation".
ACM Trans. Graph. 33, 4, Article 83 (July 2014), 10 pages
Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014
"Computational design of linkage-based characters".
ACM Trans. Graph. 33, 4, Article 64 (July 2014), 9 pages.
......

Figure 13: Suggesting Relevant RO's given a Particular Document

The DRInventor Platform offers via a single API call such functionality, by just specifying the *UUID* of the document the user wants to trigger the recommendation with. More information about this method is summarised below.

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/documents/bf0d989da2877830603a61f47f778348/
    documents
Parameters:
Method: GET
Response:
[
  "http://drinventor.eu/documents/39d9c4c0bb38b5fd740be63ad4cbb82c",
  "http://drinventor.eu/documents/fe95c24777e690d9ea8aedce1fe8610e",
  "http://drinventor.eu/documents/470c8134092ab394ee4590089add40bf",
  "http://drinventor.eu/documents/2c743b3f8d576a0145909d2f6fdca138",
  "http://drinventor.eu/documents/4638670749c720d87e6f95d3e4b91729",
  "http://drinventor.eu/documents/65401490542663e3b902ece90710e455",
  "http://drinventor.eu/documents/cda7c9724e8f8f1a9d8134c274a72900",
  "http://drinventor.eu/documents/9ee854544ced65c42674fa711510c2b9",
  "http://drinventor.eu/documents/8fdadc6929e00914a27a87b62a698355",
  "http://drinventor.eu/documents/c3160ba7a501c12efe861a3bf913bb7",
  "http://drinventor.eu/documents/2b009f85527543175a2374484fd974d7",
  "http://drinventor.eu/documents/f836450d8221e31c58915ea06b503a10",
  ...
]
```

The long list of returned documents (connections from the original document to all other documents in the corpus) can be filtered out by keeping only the top N resources and therefore adapt to the particular needs of the task and the clients performing the request. In addition, it is possible to check the similarity score between the original document (*UUID*: bf0d989da2877830603a61f47f778348) and any of those proposed by the recommendation operation (e.g. *UUID*: 65401490542663e3b902ece90710e455), by performing additional calls to the following method:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/documents/bf0d989da2877830603a61f47f778348/
    documents/65401490542663e3b902ece90710e455
Parameters:
Method: GET
Response:
{
  "uri": "http://drinventor.eu/similarities/bf0d989da2877830603a61f47f778348-65401490542663
      e3b902ece90710e455",
  "creationTime": "2016-06-28T09:31+0000",
  "weight": 0.12992112312061238
}
```

Similar information can also be accessed via the graphic Dashboard of DRInventor. When checking the information page about a particular indexed document (e.g. *Interactive Authoring of Simulation-Ready Plants*), a graph where the nodes are the most similar documents found is automatically generated as shown in Figure 14.

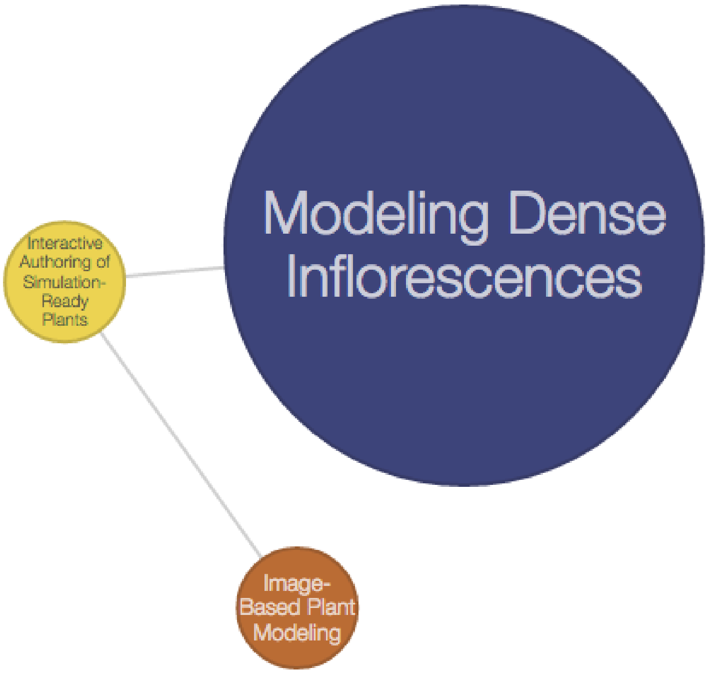

Modeling Dense Inflorescences
Interactive Authoring of Simulation-Ready Plants
Image-Based Plant Modeling

Figure 14: Most Similar Documents to *Interactive Authoring of Simulation-Ready Plants* in SIGGRAPH Corpus, and their connections

4.2 Get Relevant Resources given a Text Excerpt (Filters)

It is fairly common that a user aiming to retrieve a relevant document from the platform does not have a specific Document in mind that allows triggering the recommendation process, either because the topics he is interested in are more vague, or because he did not find a particular document fitting them. In those cases, the platform needs to provide other more flexible methods to trigger such retrieval tasks.

Having those different requirements in mind in DRInventor we have implemented an additional kind of information units called *Filters*. Those filters allow defining excerpts of plain text that the user can leverage in order to trigger recommendation operations that could provide him with relevant documents without having to provide a particular document UUID as input.

Filters can launch two kinds of recommendation operations:

- *Similarity-Based Recommendations*. Same as the previous recommendation operations in subsection 4.1, it relies on the topic-based similarity described in Section 3.3.
- *Matches*. It relies on the search capabilities of an ElasticSearch[24] instance where all the DRInventor documents have been indexed.

The differences in results provided by both methods are yet to be analysed in future exper-

[24] https://www.elastic.co/

iments, as well as to determine the adequacy of them to the different recommendation tasks that can be implemented in the scientific knowledge discovery domain. Intuitively, we can advance that recommendation tasks where the thematic of the suggested items plays an important role are better suited to the intrinsic objetives of the topic-based implementation, while the Elasticsearch-powered results can give better support to recommendations where a more strict content similarity between the related items is preferred. In addition, Elasticsearch manages very well queries based on 2 or 3 key terms, sometimes even 1. This way, the text excerpt used for triggering this variant can be significantly shorter than the one feeding the topic-based version since the longer the submitted text is, the most precise is the topic annotation over them.

Both kinds of recommendations based on Filters can be invoked by external agents via the REST API. The process is composed by two different steps: first, we create the Filter by specifying the textual fragment that will be used for triggering the search. Once the Filter is available in the platform, we can perform a second call that takes as input the text associated with that Filter and identifies the most similar resources according to the two search strategies mentioned before. In more detail, those are the calls that need be performed:

We generate a filter by specifying the plain text that we want the search to be fed with:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/filters
Parameters:
{
  "content": "Textual Excerpt to be used as the anchor that triggers the recommendation
      process",
}
Method: POST
Response:
{
  "uri": "http://drinventor.eu/filters/ce114e4501d2f4e2dcea3e17b546f339",
  "creationTime": "2016-08-29T15:17+0000",
  "content": "Textual Excerpt to be used as the anchor that triggers the recommendation
      process"
}
```

In the response of the previous call we obtain the URI of the new Filter being generated. We can now use its UUID *ce114e4501d2f4e2dcea3e17b546f339* as a parameter embedded in the URL that launches the recommendation operation. In the case of topic-based similarity recommendation, we can invoke the following API method:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/filters/ce114e4501d2f4e2dcea3e17b546f339/
    similar
Parameters:
Method: GET
Response:
[
  {
    "weight": 0.1414650755417064,
    "resource": "http://drinventor.eu/documents/8a20fec8e1c0ed7f33c697af642f6762",
    "description": "DRAPE: DRessing Any PErson"
  },
  {
    "weight": 0.1351694840332955,
    "resource": "http://drinventor.eu/documents/3cce5834df1f1ab8ce5887f1dcf180a5",
    "description": "Yarn-Level Simulation of Woven Cloth"
  },
  {
    "weight": 0.12885667712009236,
    "resource": "http://drinventor.eu/documents/b3ec66b4097285ef97c3e941c5b35378",
    "description": "Takeo Igarashi * The University of Tokyo"
  },
  {
    "weight": 0.1232186092543197,
    "resource": "http://drinventor.eu/documents/7306698585dc018f584845f8d43fa2b7",
    "description": "Example-Based Wrinkle Synthesis for Clothing Animation"
  },
...
]
```

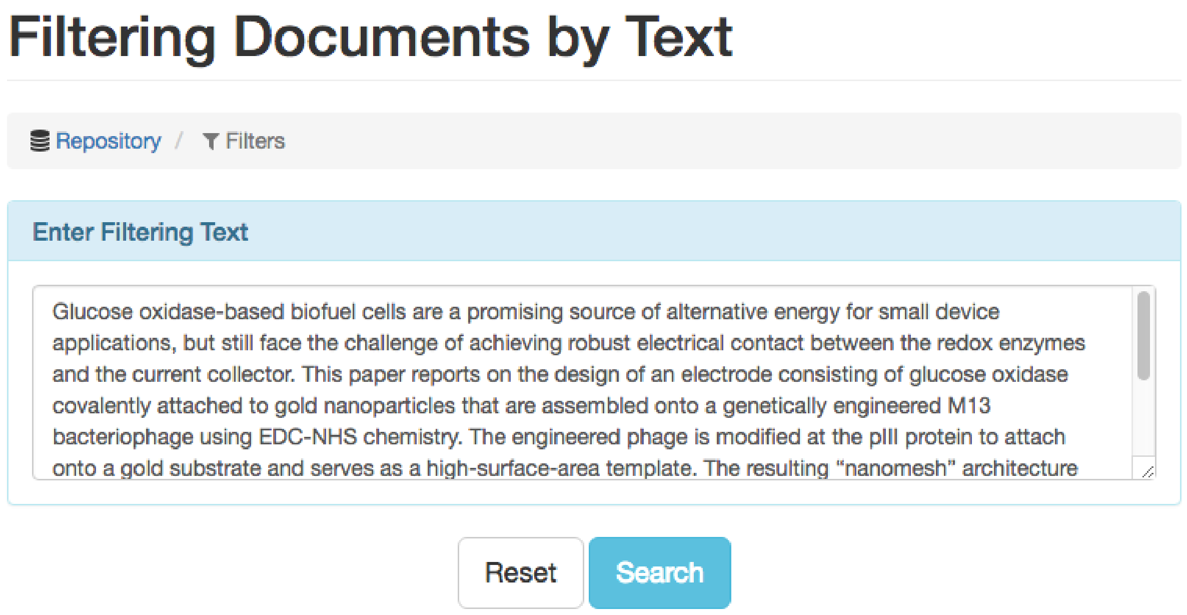
The recommendation based on the Elasticsearch capabilities (*Matches*) is also available through the dedicated API call `http://drinventor.dia.fi.upm.es:80/api/0.2/filters/ce114e4501d2f4e2dcea3e17b546f339/matches`, which accepts the same parameters than the previously described *similar* method and produces responses in the same JSON format.

The same two recommendation functionalities are exposed at the DRInventor Dashboard for a more intuitive and human friendly access to the Resource discovering capabilities. The creation of Filters is now hidden from the client, which only needs to provide the text to trigger the recommendation on the corresponding textbox (see Figure 15) and press the button *Search*, making the whole process more transparent to the users. The topic-based kind of recommendation is available under the option *Discover* → *Filters* on the DRInventor Dashboard.

The output of this topic-based recommendation operation is displayed in a table as shown in Figure 16, where the top documents suggested (according to their similarity to the text excerpt provided) are listed. Details about the authors' names, the title of the document and the similarity score with the original text are provided. You can also click on the URI of the resource in order to open a dedicated page with additional information.

The way the *Matches* kind of recommendation can be launched is essentially similar to the procedure followed for obtaining the topic-based suggestions. By clicking on the option under *Discover* → *Matches*, we access a form that allows us submiting a text excerpt (see Figure 16) and a button *Match!* for triggering the discovering process. It is important to note that the texts that you can submit for this kind of recommendation method can be shorter than the one needed for feeding topic-based recommendations, since the underlying Elastic Search index is very capable of executing searches that are composed by just a few terms, in contrast with the Topic Modelling algorithms that normally are more accurate when the input texts are longer.

The *Matches* operation of the DRInventor Dashboard can retrieve any kind of Resource in



Filtering Documents by Text

Enter Filtering Text

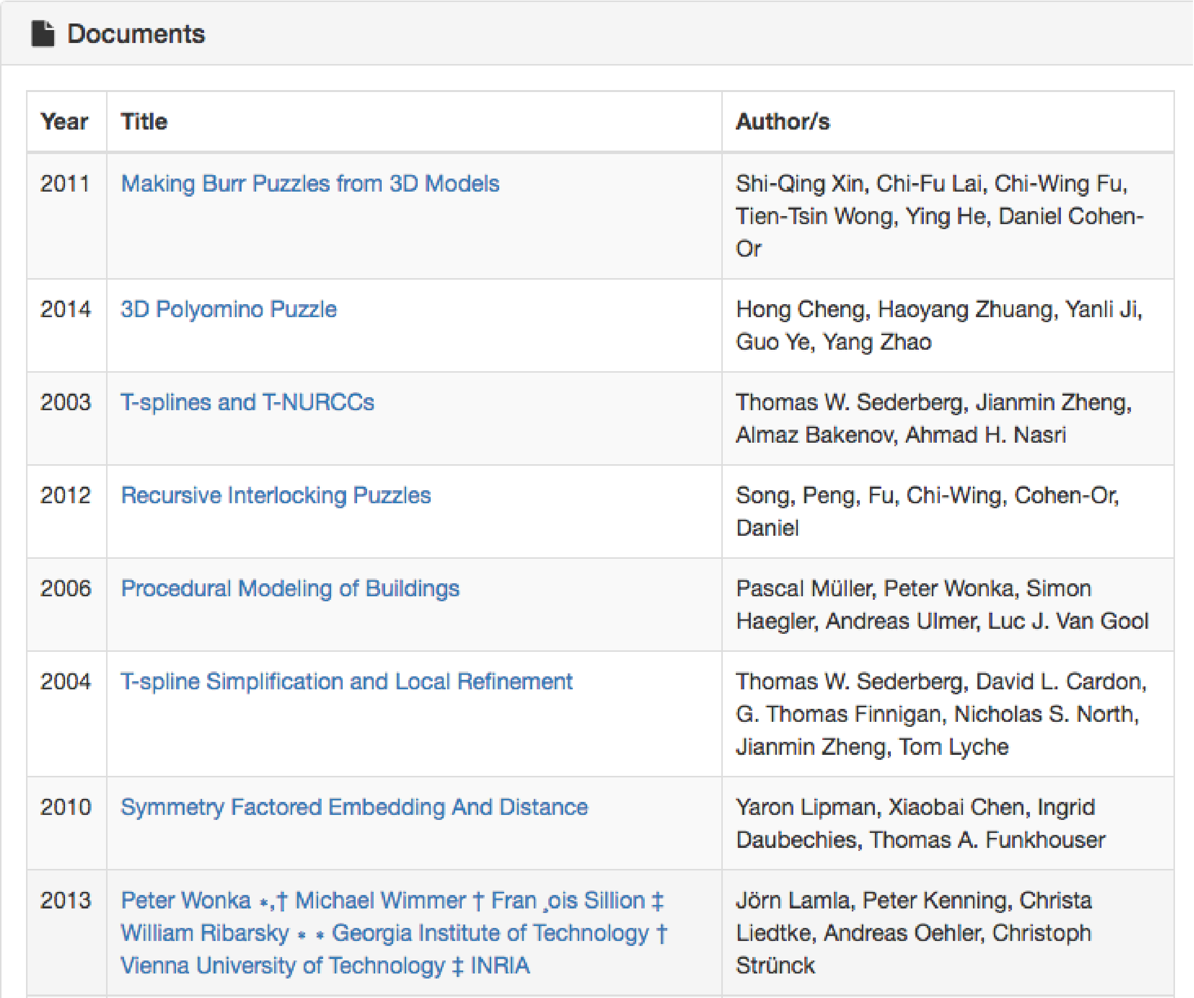
Glucose oxidase-based biofuel cells are a promising source of alternative energy for small device applications, but still face the challenge of achieving robust electrical contact between the redox enzymes and the current collector. This paper reports on the design of an electrode consisting of glucose oxidase covalently attached to gold nanoparticles that are assembled onto a genetically engineered M13 bacteriophage using EDC-NHS chemistry. The engineered phage is modified at the pIII protein to attach onto a gold substrate and serves as a high-surface-area template. The resulting "nanomesh" architecture

Figure 15: Specifying the Text Excerpt to trigger the Similar Resources Search based on LDA Topic Similarity

the platform, including Documents, Parts or Items. In the example available in Figure 18 we can see how the list of suggested resources for the input text "vertices or pixel" includes a Part and an Item as results. All resources are accompanied by a confidence score. By clicking on the URL of the resource, we can get extra information about them.

4.3 Get a Path between two Documents

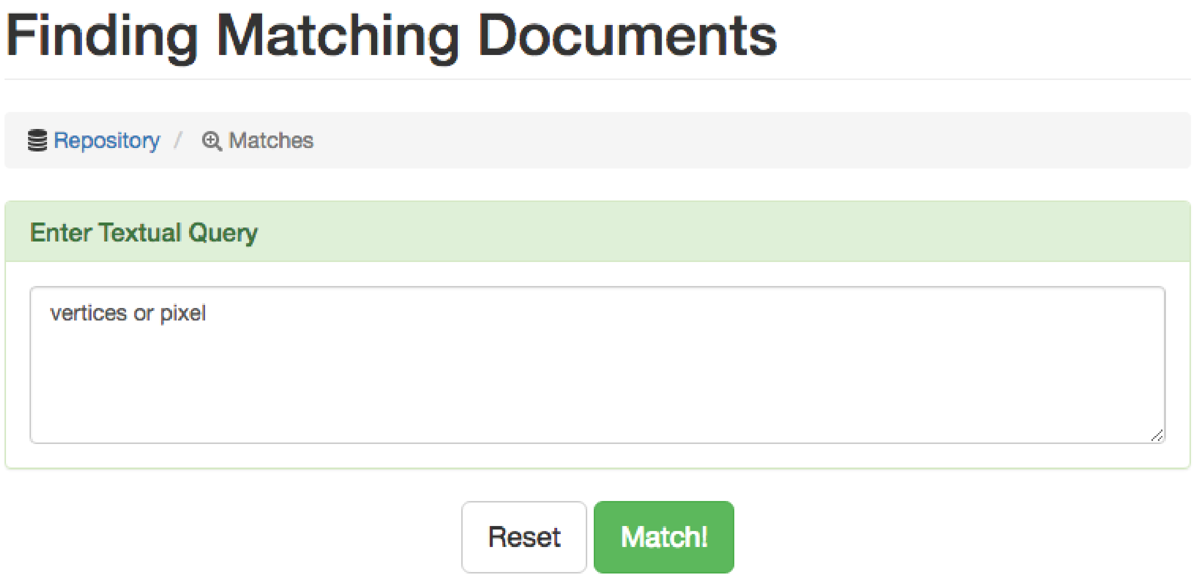
Let's imagine that we want to find possible research papers whose content overlap with the knowledge available in two different documents A and B with different thematics T_A and T_B. Those connecting papers can help us to know about research efforts that go from very T_A-oriented topics barely talking about subjects related to T_B and viceversa, to initiatives that somehow mix those two disciplines because they can be simultaneously annotated with topics T_A and T_B. Additionally, they could even include some other subjects like T_X that bring more information about why those topics can be overlapping. For example in the computer graphics domain, it would be interesting to explore how two papers (first one about lighting techniques in 3d scenarios, and the second about human faces rendering) are linked through a path of resources where those two initial topics are colliding (for example, papers on how to apply lighting on human faces in outside scenarios). This notion of a sequence of documents where a start and end documents are connected trough other consecutive resources is illustrated in Figure 19.



Documents

Year	Title	Author/s
2011	Making Burr Puzzles from 3D Models	Shi-Qing Xin, Chi-Fu Lai, Chi-Wing Fu, Tien-Tsin Wong, Ying He, Daniel Cohen-Or
2014	3D Polyomino Puzzle	Hong Cheng, Haoyang Zhuang, Yanli Ji, Guo Ye, Yang Zhao
2003	T-splines and T-NURCCs	Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, Ahmad H. Nasri
2012	Recursive Interlocking Puzzles	Song, Peng, Fu, Chi-Wing, Cohen-Or, Daniel
2006	Procedural Modeling of Buildings	Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, Luc J. Van Gool
2004	T-spline Simplification and Local Refinement	Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, Tom Lyche
2010	Symmetry Factored Embedding And Distance	Yaron Lipman, Xiaobai Chen, Ingrid Daubechies, Thomas A. Funkhouser
2013	Peter Wonka ∗,† Michael Wimmer † Fran ¸ois Sillion ‡ William Ribarsky ∗ ∗ Georgia Institute of Technology † Vienna University of Technology ‡ INRIA	Jörn Lamla, Peter Kenning, Christa Liedtke, Andreas Oehler, Christoph Strünck

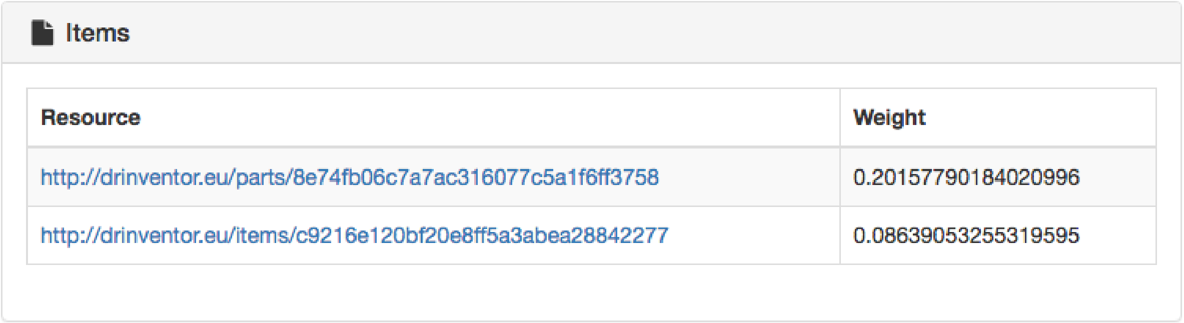
Figure 16: Resulting Documents from the Similar Resources Search based on LDA Topic Similarity for the Previously Specified Input



Finding Matching Documents

vertices or pixel

Figure 17: Specifying the Text Excerpt to trigger the Similar Resources Search based on ElasticSearch



Resource	Weight
http://drinventor.eu/parts/8e74fb06c7a7ac316077c5a1f6ff3758	0.20157790184020996
http://drinventor.eu/items/c9216e120bf20e8ff5a3abea28842277	0.08639053255319595

Figure 18: Resulting Resources from the Similar Resources Search based on ElasticSearch for the Previously Specified Input

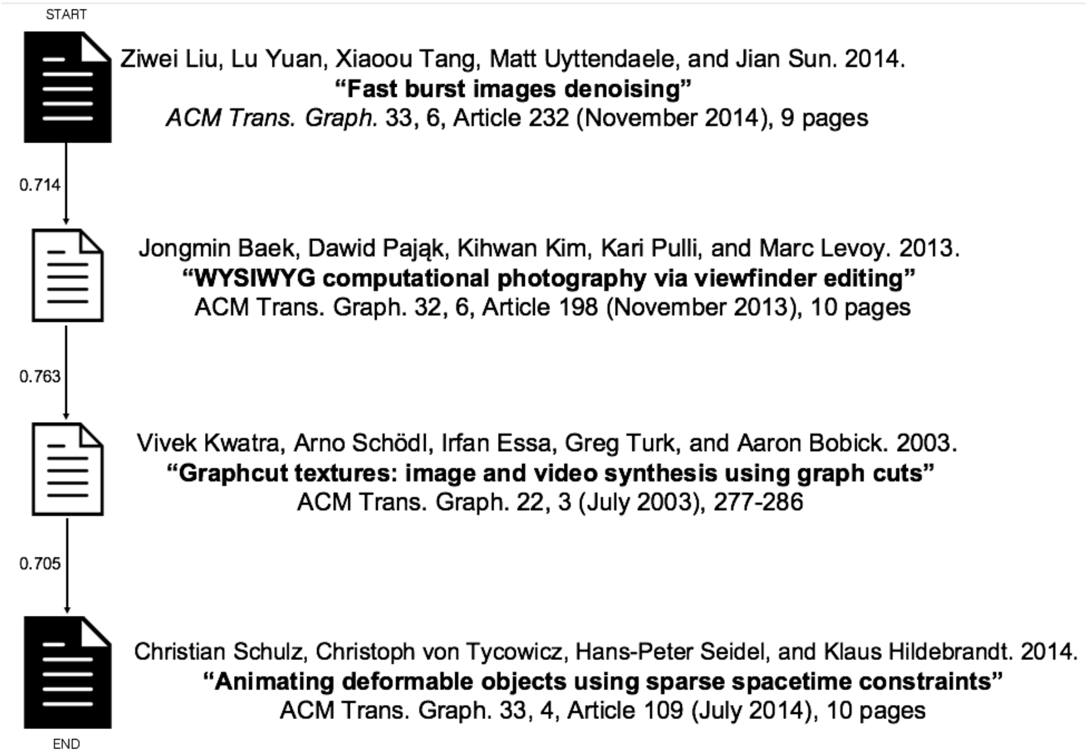

START
Ziwei Liu, Lu Yuan, Xiaoou Tang, Matt Uyttendaele, and Jian Sun. 2014.
"Fast burst images denoising"
ACM Trans. Graph. 33, 6, Article 232 (November 2014), 9 pages
0.714
Jongmin Baek, Dawid Pająk, Kihwan Kim, Kari Pulli, and Marc Levoy. 2013.
"WYSIWYG computational photography via viewfinder editing"
ACM Trans. Graph. 32, 6, Article 198 (November 2013), 10 pages
0.763
Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003.
"Graphcut textures: image and video synthesis using graph cuts"
ACM Trans. Graph. 22, 3 (July 2003), 277-286
0.705
Christian Schulz, Christoph von Tycowicz, Hans-Peter Seidel, and Klaus Hildebrandt. 2014.
"Animating deformable objects using sparse spacetime constraints"
ACM Trans. Graph. 33, 4, Article 109 (July 2014), 10 pages
END

Figure 19: Suggesting Relevant RO's given a Particular Document

The set of Resources in the DRInventor Platform, and the topic-based similarity relations that are calculated over them as explained in previous Section 3.3 shape up a graph that can be traversed in order to find relevant paths between resources that are not directly connected beforehand. DRInventor is able to identify those paths by applying path finding algorithms (in particular, the well-known $A*$ approach) in order to efficiently find traverse paths between pairs of nodes in the aforementioned graph.

Such functionality is exposed via both the API and the Dashboard. As it was the case with the recommendations (see previous Section 4.2), the programmatic access to this feature through the API is divided in two different REST methods. The first one is intended to create a resource of type Path inside the platform. In order to do so, we need to specify the *Origin* document, and the *Destination*. Having as example an origin document with UUID *d1a2239e6a19eea1780b051b0c68cb02* and a destination document with UUID *bb108cdfffcfb020aa946784be* we should invoke the following API method:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/paths
Parameters:
{
  "end": "d1a2239e6a19eea1780b051b0c68cb02",
  "start": "bb108cdfffcfb020aa946784be7e6e2a"
}
Method: POST
Response:
{
  "uri": "http://drinventor.eu/paths/443e7c322c6324f7e763fe28901d2a5e",
  "creationTime": "2016-08-31T17:25+0000",
  "start": "bb108cdfffcfb020aa946784be7e6e2a",
  "end": "d1a2239e6a19eea1780b051b0c68cb02"
}
```

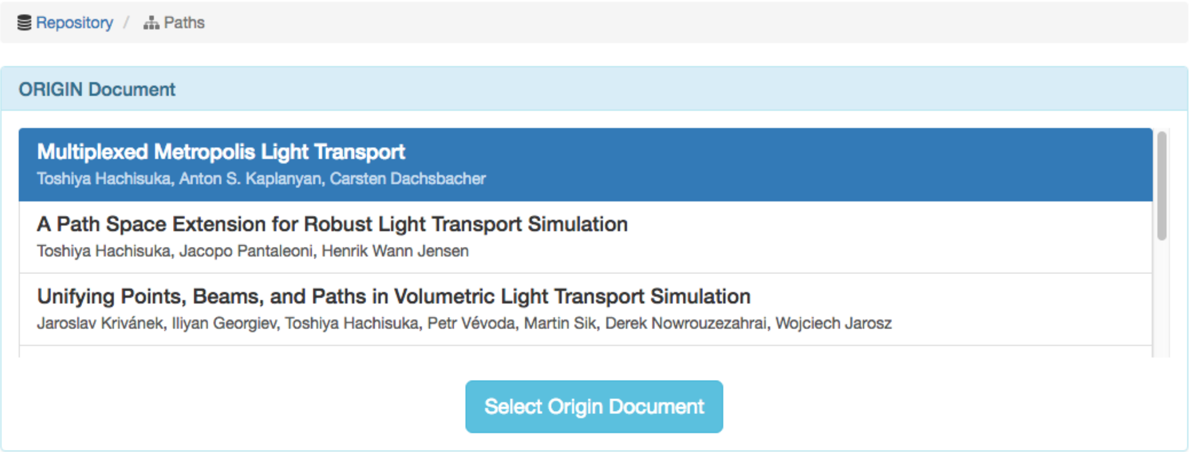
The second step consists of requesting to the platform the calculation of a relevant set of connected resources linking the origin and destination documents. In order to do so, and having obtained from the previous API call the UUID of the Path we has just created (in the example, *443e7c322c6324f7e763fe28901d2a5e*), we can launch the pathfinding algorithm by relying on the method below:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/paths/443e7c322c6324f7e763fe28901d2a5e/
    documents
Parameters:
Method: GET
Response:
[
  {
    "weight": 0,
    "resource": "http://drinventor.eu/documents/39d9c4c0bb38b5fd740be63ad4cbb82c",
    "description": "Fast Burst Images Denoising"
  },
  {
    "weight": 0.7145796859089336,
    "resource": "http://drinventor.eu/documents/3096381661bdc152f2c1af913ddb522b",
    "description": "WYSIWYG Computational Photography via Viewfinder Editing"
  },
  {
    "weight": 0.7634335807003932,
    "resource": "http://drinventor.eu/documents/8a42243e4e1b4bb4f91862aee1d2d6ad",
    "description": "Graphcut Textures: Image and Video Synthesis Using Graph Cuts"
  },
  {
    "weight": 0.7053453460827446,
    "resource": "http://drinventor.eu/documents/470c8134092ab394ee4590089add40bf",
    "description": "Animating Deformable Objects using Sparse Spacetime Constraints"
  }
]
```

This API call is designed to retrieve paths of Documents as specified in the final part of the URL (*/documents*). We can also request paths composed by other kind of resources by specifying their names at the end, such as */items* or */parts*.

This functionality has been also included in the dashboard under the option *Discover* → *Paths*. Again here, the process of generating paths between two resources becomes more transparent than in the API since the creation of the intermediate clase Path for keeping track of the origin and destination documents is internally performed by the prototype without any further intervention from the user.

The first step consists of selecting the origin document that indicates the research work used as starting point for building the path. For demo purposes and in order to keep the interaction as simple as possible, the prototype contains a pre-defined list of research documents in order to pick up the original document from (see Figure 20).



Repository / Paths

ORIGIN Document

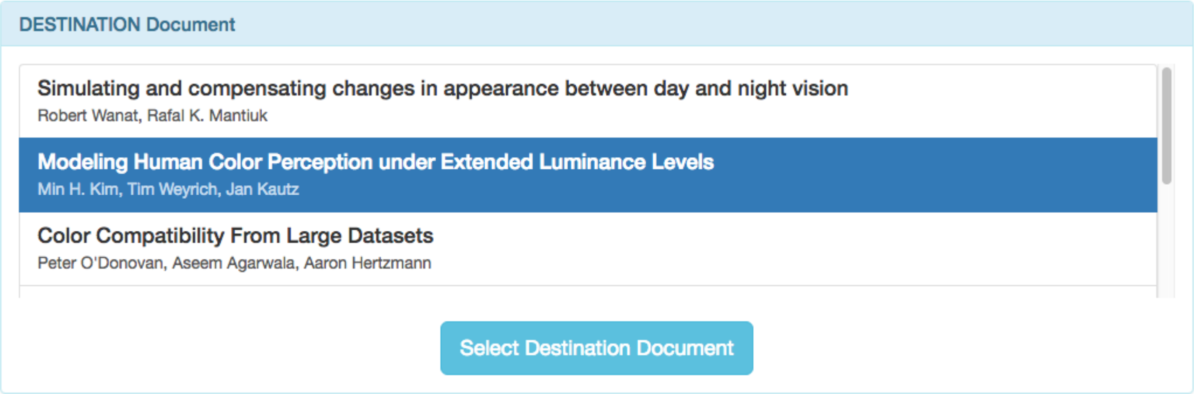
Multiplexed Metropolis Light Transport
Toshiya Hachisuka, Anton S. Kaplanyan, Carsten Dachsbacher

A Path Space Extension for Robust Light Transport Simulation
Toshiya Hachisuka, Jacopo Pantaleoni, Henrik Wann Jensen

Unifying Points, Beams, and Paths in Volumetric Light Transport Simulation
Jaroslav Krivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Sik, Derek Nowrouzezahrai, Wojciech Jarosz

Figure 20: Specifying the Document to be considered as the Origin of the Resources Path

Right after the “Select Origin Document” button is pressed, the prototype displays a new list of documents from where we can pick up the destination document by following the same mechanism relying on a set of pre-selected documents as shown in Figure 21. Once we have made our choice, we click on the button “Select Destination Document” and the process of internally generating the Path is launched.



DESTINATION Document

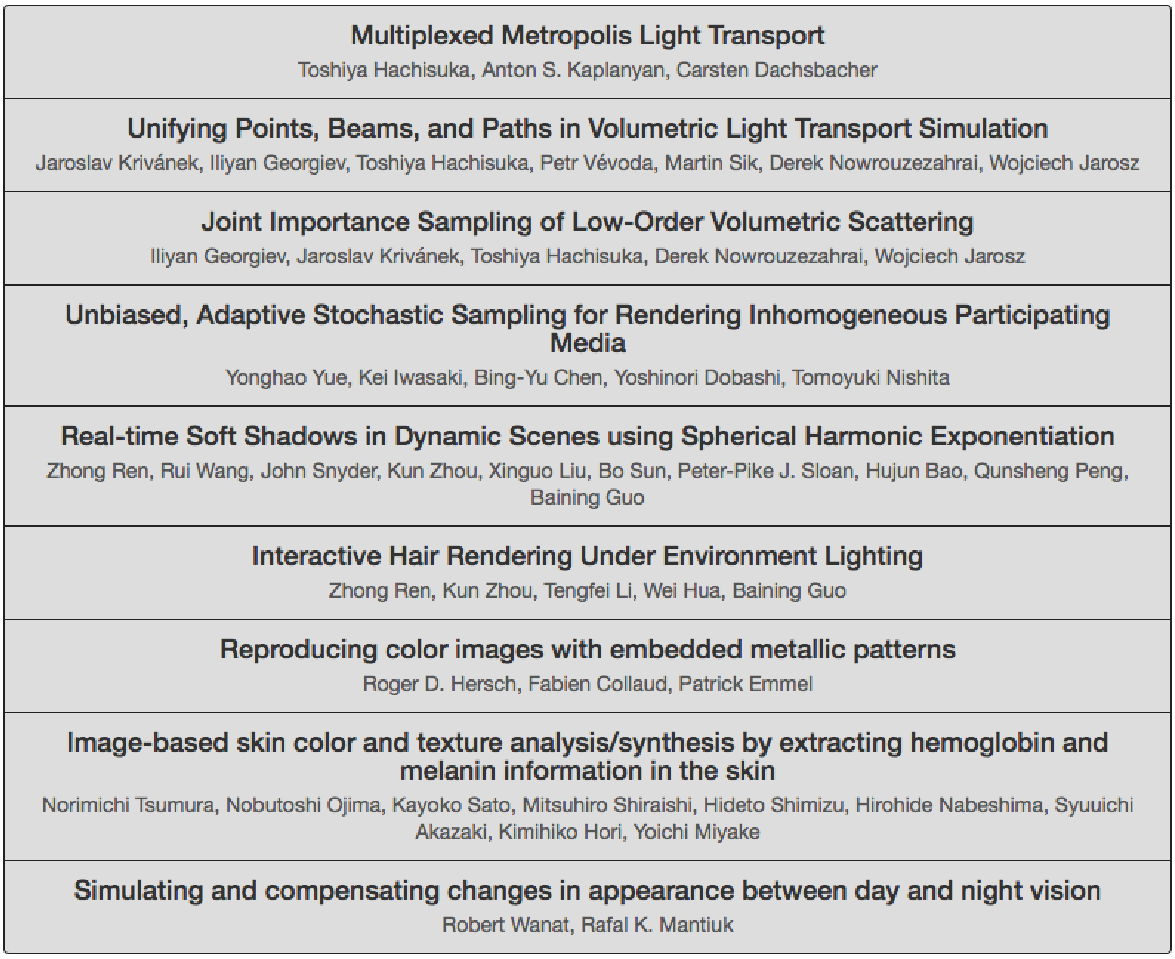
Simulating and compensating changes in appearance between day and night vision
Robert Wanat, Rafal K. Mantiuk

Modeling Human Color Perception under Extended Luminance Levels
Min H. Kim, Tim Weyrich, Jan Kautz

Color Compatibility From Large Datasets
Peter O'Donovan, Aseem Agarwala, Aaron Hertzmann

Figure 21: Specifying the Document to be considered as the Destination of the Resources Path

Once the shortest path calculations over the graph of documents and their similarity distances is finished, the list of intermediate documents that sequentially allow us to move from the thematics in the origin documents to the ones in the destination is revealed to the client in the particular order they have to be traversed. We can see an example of such list in Figure 22



Multiplexed Metropolis Light Transport
Toshiya Hachisuka, Anton S. Kaplanyan, Carsten Dachsbacher

Unifying Points, Beams, and Paths in Volumetric Light Transport Simulation
Jaroslav Křivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Sik, Derek Nowrouzezahrai, Wojciech Jarosz

Joint Importance Sampling of Low-Order Volumetric Scattering
Iliyan Georgiev, Jaroslav Křivánek, Toshiya Hachisuka, Derek Nowrouzezahrai, Wojciech Jarosz

Unbiased, Adaptive Stochastic Sampling for Rendering Inhomogeneous Participating Media
Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, Tomoyuki Nishita

Real-time Soft Shadows in Dynamic Scenes using Spherical Harmonic Exponentiation
Zhong Ren, Rui Wang, John Snyder, Kun Zhou, Xinguo Liu, Bo Sun, Peter-Pike J. Sloan, Hujun Bao, Qunsheng Peng, Baining Guo

Interactive Hair Rendering Under Environment Lighting
Zhong Ren, Kun Zhou, Tengfei Li, Wei Hua, Baining Guo

Reproducing color images with embedded metallic patterns
Roger D. Hersch, Fabien Collaud, Patrick Emmel

Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin
Norimichi Tsumura, Nobutoshi Ojima, Kayoko Sato, Mitsuhiro Shiraishi, Hideto Shimizu, Hirohide Nabeshima, Syuuichi Akazaki, Kimihiko Hori, Yoichi Miyake

Simulating and compensating changes in appearance between day and night vision
Robert Wanat, Rafal K. Mantiuk

Figure 22: List of Resources inside the Platform linking the Origin Document with the Destination

4.4 Cluster Documents according to Topics

The huge volume of documents some corpuses contain makes necessary to establish mechanisms and techniques that help humans to glance the big picture of the information contained in them, so they can quickly know about the different domains represented in them. At the same time, knowing those relevant topics beforehand can improve the quality of the operations we launch over the data, allowing the users to have a better idea about what kind of results they can be able to obtain.

For example, in our use case based on the SIGGRAPH dataset, we know beforehand that papers are about *Computer Graphichs*, but this is a very wide discipline. Hence it becomes important to access in a quick transparent matter to the set of subtopics that are contained in the corpus, for example: *Plant Redering*, *Raytracing* or *Object tracking*. It would be also interesting to know how they are distributed inside the dataset in terms of number of documents where each subject is more prominent. This way we would be able to identify those subjects that are highly covered, against those which are barely present in just a few research objects.

This operation is also available in the DRInventor Platform by accessing the option *Explore* → *Corpus*, which was partially described in Section 2.5.1. This page offers different advanced visualisations of the indexed papers and their annotations, like a graph of prominent topics. In Figure 23 we can visualise the set of documents in the platform, where the similarity scores calculated between pairs are used to generate a 3d cloud of nodes and arcs between them (left-side plot) or place them over a 2D plane (right-side plot). In addition, certain parts of the plots have been coloured according to the 7 topics found on the collection (see information on why this number of topic has been used in Section 3.2), providing a visual idea of how the set of documents is clustered into different groups with similar thematics and therefore offering that high level view of the topics contained in the corpora and their distribution.

4.5 Other Advanced Operations in the DRInventor Platform

The huge amount of information that is becoming available today for experts and users in the domains turns the task of making sense out of all this data into a much more complicated process that normally requires human intervention and forces the consumers to perform significant efforts in order to exploit knowledge in an efficient way.

In this last subsection we include a set of innovative recommendation operations that complement the previously described features to give better idea of the possibilities that those semantically-driven recommendations are intended to bring into the table for improving the way the research objects can be proposed to the users.

The operations described below intend to complement the previously described features in order to offer a better understanding about the promising advantages that those techniques are capable of offering. Even they are not directly exposed via API or implemented in the DRInventor Framework, *they can be implemented over the data and the functionalities already*

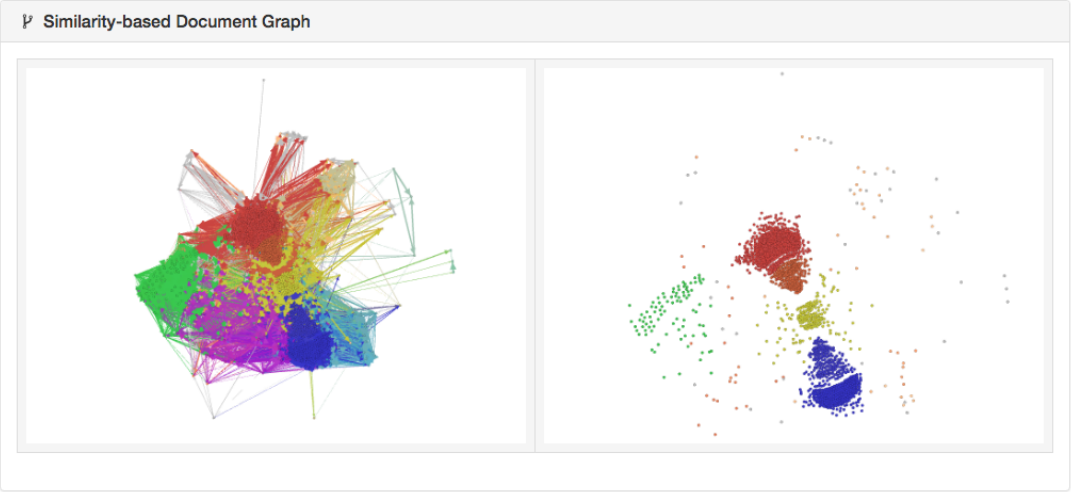


Figure 23: Resources from SIGGRAPG corpus indexed in DRInventor Platform and Clustered according to Most Prominent Topics

considered and in the platform.

4.5.1 Generating Knowledge Routes

The goal of this feature is to find a list of resources in the corpus connecting two research areas. This operation has many commonalities with the path generation method described in Section 4.3, but now both initial and ending anchors are general topics and not specific documents in the platform.

This kind of operation can bring a lot of value to the community: for example, an author having knowledge about some area, for instance Computer Graphics, wants to explore another area such as Astrophysics in order to identify papers or ROs that may be relevant to him/her. From a classical point of view, the next step for the author would be to read papers and/or books about Astrophysics to gain more knowledge about this area but this could be quite hard. In order to overcome this "cold start" problem, the DRInventor Platform can offer a soft transition between these areas showing a sorted list of the most representative ROs that connect them, allowing authors to get to know the final knowledge by incrementally reading more topic-specific papers.

This is possible thanks to the undirected graph to model pairwise similarity relations between ROs that is built into the DRInventor Platform. It is based on the previously mentioned similarity measures (Section 3.3). External agents can implement applications that read the words or concepts defined by the users, describing the starting and the ending research areas. Then, the system obtains the topics distributions where these words or concepts have higher probabilities. For each of these topic distributions, the system chooses the most similar ROs that are based on the distance measure defined in Section 5. Now, using the Dijkstra's

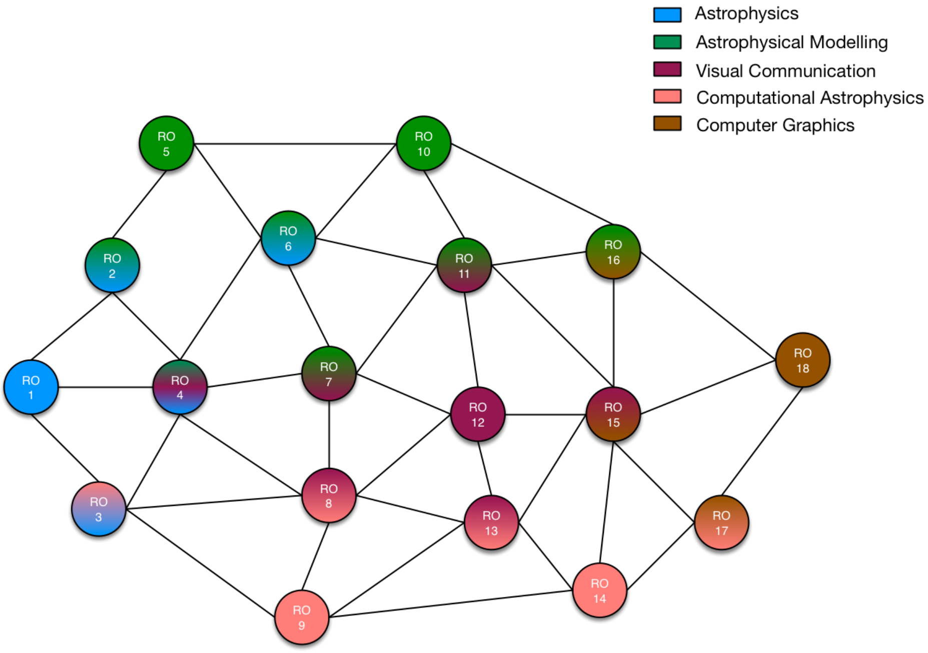


Figure 24: Research Object-Graph sample

inspired [14] A^* algorithm, the system will obtain the minimum cost path between RO's in the graph. The route will be presented to the author as a sorted list of ROs to gain the desired knowledge.

Continuing with the example, the author will receive a sorted list of ROs to pass from Computer Graphics to Astrophysics, being the first RO the most representative research object in Computer Graphics (based on words/concepts provided by the author) and the last one the most representative research object in Astrophysics. The rest of ROs in the chain depend on the similarity measure used to connect the ROs in the graph. We can also think about a more fine grained recommendation operation where the author may choose a *direct*, a *uniform*, or a *balanced route-of-knowledge*.

- ***direct***: Based exclusively on the *content* of ROs, this solution takes into account the content-based similarity measure under topics distribution described in Section 3.3.1 to only connect the ROs of similar *research areas*. For example, RO_1, RO_3, RO_9, RO_{14}, RO_{17} and RO_{18} (*Figure* 24).

- ***uniform***: Based exclusively on the *context* of ROs, this solution takes into account the content-based similarity measure under topics distribution described in Section 3.3.2 to only connect the ROs built in a *similar way*. At present, the system prioritizes ROs with similar authors, but future work will incorporate other aspects such as style of writing, aggregated resources, etc. For example, RO_1, RO_2, RO_5, RO_6, RO_{10}, RO_{11}, RO_{12}, RO_{13}, RO_{15}, RO_{16} and RO_{18} (*Figure* 24).

– ***balanced***: Based on both the *content and context* of ROs, this solution takes into account the similarity measure defined in Section 2, maintaining the appropriate balance between what is the research topic, which researches are involved and how it was built. This solution can be considered more complete and finely tuned than previous ones, but really any of them are good options. For example, RO_1, RO_4, RO_6, RO_7, RO_{11}, RO_{15} and RO_{18} (*Figure* 24).

Moreover, an author may filter ROs according to *publishing date*, *license rights*, *format*, etc. In those cases the system will provide a better suited subgraph including only the nodes that verify those criteria.

4.5.2 Towards a Linked-Research

Each of topics discovered in the LDA model describes a research area by its most relevant concepts or words. Moreover, research objects have a topic distribution assigned, so the meta information associated to each resource in the platform can be crossed with the previous research areas discovered in other corpora with initially different purposes, such as research centres or laboratories where the works were carried out, details about the conferences where the research objects were published such as dates of publication or acceptance rate, and even public information about authors concerning institutions where they have performed their activities.

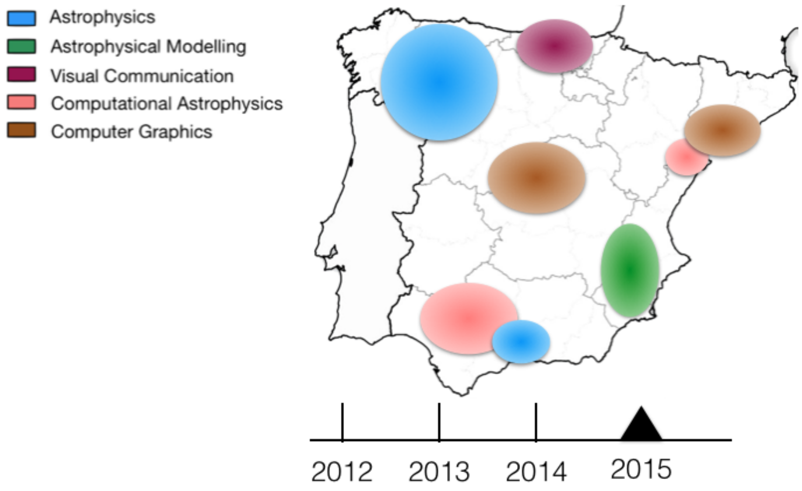
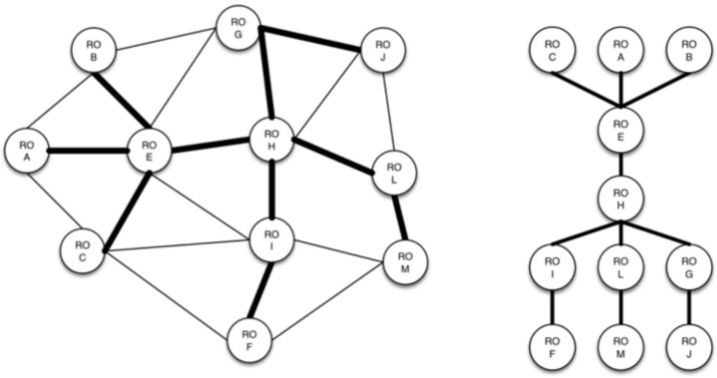


Figure 25: Simulation of some research topic locations in Spain during 2015

By exposing all this information in a *Linked Data* fashion, connected with other Web Resources and making accessible their particular properties, the system can produce additional knowledge such as the regions in a country where more publications about a specific research area have been done during a period of time, or funds granted for a type of research projects.



(a) Figure A

(b) Figure B

Figure 26: Optimal review of research objects

For instance, in the figure 25 a map of Spain is shown indicating a potential distribution of some research areas (Astrophysics, Astrophysical Modelling, Visual Communication, Computational Astrophysics and Computer Graphics) in that country based on the research center where the authors work at the time of publication. This could be useful for an author who wants to know how trendy is a research field or where is the best place to develop or discuss an idea, or who is looking for funds or grants to complete the research she/he is currently tackling. The relations established between those different sources can boost the discovering of hidden details about what is being research, where, when, and by whom.

4.5.3 Defining an Optimal Paper Review Process

The goal of this recommendation operation is to find the best way to read a group of research objects. Let's suppose that an author has marked a group of research objects as interesting for her/his work and she/he needs to read them. As showed in Figure 26, these research objects are A, B, C, E, F, G, H, I, J, L and M. So, she/he can read them in a random order (Figure 26(a)) or let DRInventor Platform to define an order based on similarities between resources that are automatically generated between them (Figure 26(b)). This recommendation is somehow similar to the path calculation feature described in Section 4.3 and therefore also to the *Route-of-Knowledge* use case described above. The difference lies in the fact that now only the research objects provided can be used to build up the path, and none of them can be omitted. As introduced before and according to the similarity dimensions we have considered in our implementation, the system would be able to offer a *direct*, *consistent* or *balanced* solution as aforementioned.

5 Conclusions

In this deliverable we have presented a set of recommendation features implemented over the DRInventor Platform in order to ease the way scientists and researchers are provided with relevant information about a particular domain they are interested on. Through the implementation of novel annotation techniques such as ontology learning or probabilistic topic modelling, we are able to automatically make sense out of big collection of documents, identify connections between similar documents and arrange them in groups according to the most prominent subjects they cover.

The repository of Research Objects indexed in DRInventor Platform innovates the way scientific papers in particular and textual resources in general are stored and managed, by considering different units of information ranging from entire documents, to specific items inside them or parts (abstract, introduction, conclusion...). Therefore the recommendation features implemented are able to deal with this higher level of granularity to better suit to the data consumers' needs.

Between the most remarkable set of recommendation operations implemented there are some functionalities for clustering the corpus of documents into set of resources according to their topics, methods for suggesting resources that are relevant to a given an except of text given by the user, and a logic for generating sequences of topic-wise similar resources that can be followed to transit from the knowledge expressed in a document to the one available in another.

All the resources, annotations and the basic logic for implementing the described features have been made available through an API. The Librairy Toolbox includes as well many of the functionalities described so they can be replicated in similar text-oriented scenarios without extra effort. The techniques implemented open a window to a new set of information consuming assistants with novel features that we will introduce in the next and last Section 6, helping different users and experts in the domain to make the most of their tasks by releasing them from dealing with the burden of textual documents that are becoming available everywhere.

6 Future Work

The work on automatic knowledge discovery and recommendations that has been described in this deliverable brings into the table a new research scenario where scientist are efficiently assisted by Information Extraction, Knowledge Representation, and Artificial Intelligent methods that intend to ease the tasks they perform on a daily basis (searching for new relevant papers on a particular matter, detecting new groundbreaking research being developed, etc) or improve their outcomes.

The features presented in the DRInventor Platform allow taking into account with a much higher set of documents that are processed in considerably shorter periods of time, and leverage on advanced techniques producing insights that are closer to what the experts in the domain would have suggested if they had infinite time to manually process the entire corpora.

However and keeping in mind the great value that these solutions are already offering, there are still various aspects that need to be further investigated, optimised or developed. Given the cutting-edge nature of this domain of research the number of future lines that need to remain open is quite significant. In this Section we introduce some of the most remarkable ones.

6.1 Short Term Developments in DRInventor Platform

Keep Developing the DRInventor Dashboard. The current version of the dashboard allows to easily navigate resources and trigger some recommendations applications. However some of the functionalities described in this deliverable are not exposed there so clients wanting to test them have to implement their own prototypes by leveraging on the API. In order to quickly show the potential of this platform to any potential consumer without forcing him to implement its own prototype, we plan to keep working on developing innovative GUI's showcasing the advanced operations offered by the system.

Investigate the Behaviour of the Platform when Indexing RO Datasets from other domains. The different use cases in this deliverable have been designed to be performed over the SIGGRAGH corpus, which is completely focused on the Computer Graphics domain. It would be very interesting to extrapolate the different advantages reported over the different sections to a different domain like *Web Semantic* to see if they keep probing valid over a completely different research dataset. We can find repositories where those new RO's can be found by resorting to some catalogs offered by important journals in the field, such as the Semantic Web Journal[25]. Once those external endpoints have been selected, the ingestion, indexing and annotation processes are straightforward to be applied, making this future effort very plausible to be achieved without any considerable human and temporal costs.

[25] http://www.semantic-web-journal.net/ReviewedAccepted

6.2 Boosting the Performance of Implemented Techniques

Better Exploit Terms and Implement a New Version of the Learning Algorithm. The current version of the learning algorithm implements a very naive algorithm that has been described in [4] and needs a major revision and optimisation. Given this fact, the Terms contained in the Platform have not been sufficiently exploited despite their crucial importance to automatically construct the underlying model describing the of the corpus, and offer such a strong conceptual representation to better contextualise all the recommendation operations.

Consider More Dimensions during the Calculation of Similarities Between Documents. At the moment, only contextual details (metadata about the resources, like author or publication dates), together with the topic modelling results are considered to decide on how much two resources are distant. In order to improve the precision of such operation, we can rely on additional features that can be combined together to offer better insights about those similarities. For example, following up the previous point on improving the learning algorithm, we can rely on the frequency of relevant terms available in the resources being compared. We can also consider how similar those detected concepts are by leveraging on semantic distances implemented over specific or general knowledge bases like Babelnet[26] or DBpedia[27].

Better Exploit the Parts of the Scientific Discourse. DRInventor Platform is able to detect different part of the scientific discourse (Abstract, Introduction, Conclusions...) and annotate them accordingly. However, when calculating topics and distances over resources, this information is not considered. We can suppose that some of those parts (for example, the conclusions) are more adequate to be exploited when performing particular recommendations, given they capture particular aspects of the documents that are more in line with the objective of the information retrieval task. Being able to identify which parts are more suitable for each discovery process and therefore better leveraging on the annotations generated from those excerpts is a research line that can produce significant improvements on the implemented features.

6.3 Evaluating the Implemented Techniques

Closely related with the previous point, the best way to quantitatively measuring the improvement in the techniques through the different versions of the algorithms used for annotating resources and offering advanced features over them is to evaluate them in a formal way. In Deliverable 5.5 of DRInventor Project [22] we target how to perform a first evaluation of the Terms generated by the ontology learning module. Similar efforts need to be made for checking the adequacy of the topics generated, and in the correctness of the similarity scores that are obtained by relying on them.

[26] `http://babelnet.org/`

[27] `http://wiki.dbpedia.org/`

6.4 Implement the Notion of Users Using the the Platform

Probably the most significant future research line to be tackled is the implementation of the concept of users inside the platform (in our case, scientists). Recommendations can be hugely benefited from exploiting information about the previous behaviour of that same person or other individuals which potentially share preferences and research interests. We need different methods for storing and tracking the most relevant features defining those users, and be able to combine it in a smart way with the advanced annotations of the content they access so recommendations can exploit then conveniently for offering more tailored suggestions of research objects to be consumed.

We can also think about importing Authors from external repositories, like we already do with research objects. For example we can update the system to handle digital identifications such as ORCID[28], which provide a persistent digital identifier that distinguishes an author from every other researcher and supports automated linkages to external services such as Scopus[29], ResearcherID[30] or LinkedIn[31], allowing to get information about her/his professional activities and therefore be able to exploit details about the user collected from external platforms.

Finally we show a couple of examples of what kind of recommendation operations would be possible to achieve if we had users in the platform:

Next-Step: finding the next research area based on the historical publications of an author.By leveraging all ROs that an author has published or read, the system is able to predict the topic distribution of the next research area he should tackle. The system create *trending vectors* splitting the topic distribution of each RO by topics. For each *trending vector* a linear regression is calculated to know the probability of that topic to become the next research according to the historical information. The system will show to the author a list of ROs related to that distribution of topics to facilitate a first exploration.

Besides, similar to the mutation operator in genetic algorithms, the system can introduce a random modification to the distribution of topics to propose new research areas not too far from the current research line showing some statistics about them such as number of ROs published, date of the last one (*hot-topic*), research centres specialised in that field, and so on.

Future-Collaborations: Finding a relevant list of authors to collaborate with

Inspired by the graph of resources linked by their similarities scores, the system can also create an *author-graph* connecting authors whose *author-similarity* value is high enough. The similarity between two authors can be calculated upon the similarity in their publications and the associated features stored in the platform or retrieved from external sources, such as the affiliation or the H-Index.

[28] http://orcid.org
[29] http://www.scopus.com
[30] http://www.researcherid.com
[31] https://www.linkedin.com

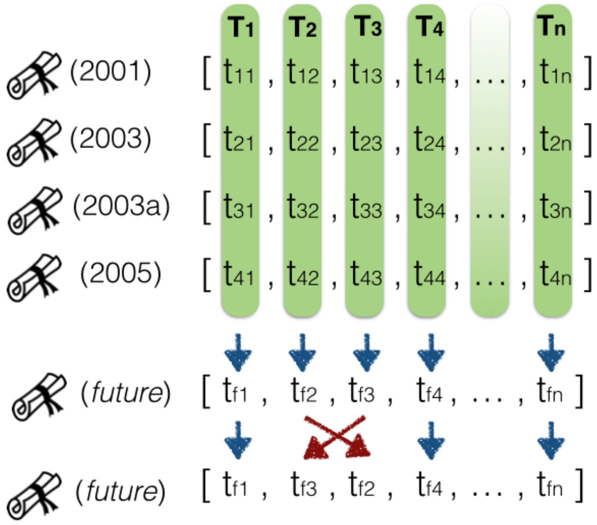
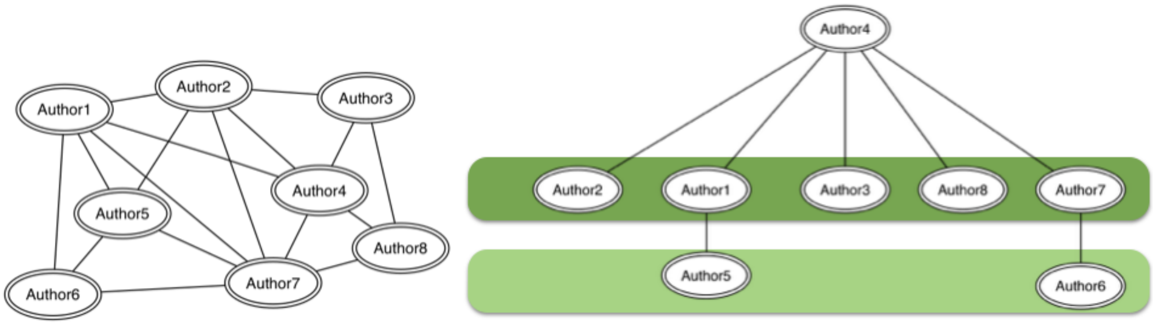

T1
T2
T3
T4
Tn
(2001) [t11 , t12 , t13 , t14 , … , t1n]
(2003) [t21 , t22 , t23 , t24 , … , t2n]
(2003a) [t31 , t32 , t33 , t34 , … , t3n]
(2005) [t41 , t42 , t43 , t44 , … , t4n]
(future) [tf1 , tf2 , tf3 , tf4 , … , tfn]
(future) [tf1 , tf3 , tf2 , tf4 , … , tfn]

Figure 27: Trending vectors from publications of an author



(a) *authors-graph*

(b) *authors-tree*

Figure 28: Graph and Tree of authors to obtain future collaborations

Thus considering that we have the author-graph shown in Figure 28(a), and supposing we are interested to know what other authors can be relevant for author number 4 based on their publications and context, we'll discover that mainly the authors 2, 1, 3, 8 and 7, and then authors 5 and 6 may be interesting for him according to the generated graph.

References

[1] D. Allard, a. Comunian, and P. Renard. Probability Aggregation Methods in Geoscience. *Mathematical Geosciences*, 44(5):545–581, 2012.

[2] Apache. Apache Spark. *https://spark.apache.org/docs/latest/mllib-clustering.html#latent-dirichlet-allocation-lda*, 2015.

[3] Arthur Asuncion, Max Welling, Padraic Smyth, and Yee Whye Teh. On Smoothing and Inference for Topic Models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34, 2009.

[4] Carlos Badenes, Rafael Gonzalez, Oscar Corcho, and Feng Dong. Repository of indexed ros. Technical report, 2015.

[5] Frederick Betz. Managing Science. *Knowledge Management*, page 190, 2011.

[6] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

[8] a Celikyilmaz, D Hakkani-Tur, and Gokhan Tur. LDA Based Similarity Modeling for Question Answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, pages 1–9, 2010.

[9] Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine Learning*, 34(1-3):43–69, 1999.

[10] Kalyanmoy Deb and Ram Bhushan Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9:1–34, 1994.

[11] Kalyanmoy Deb and Mayank Goyal. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics*, 26(1):30–45, 1996.

[12] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. 18(c):1–1, 2013.

[13] Kalyanmoy Deb and Santosh Tiwari. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3):1062–1087, 2008.

[14] Edsger Wybe Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[15] Marti A. Hearst. Untangling text data mining. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10, 1999.

[16] Thomas Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[17] Khaos Investigacion. JMetal. *http://jmetal.sourceforge.net*, 2015.

[18] K Liagkouras and K Metaxiotis. An Elitist Polynomial Mutation Operator for improved performance of MOEAs in Computer Networks. *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pages 1–5, 2013.

[19] Francesco Osborne, Giuseppe Scavo, and Enrico Motta. Identifying Diachronic Topic-Based Research Communities by Clustering Shared Research Trajectories. *Lecture Notes in Computer Science*, 8465:114–129, 2014.

[20] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, 2007.

[21] Roopesh Ranjan and Tilmann Gneiting. Combining probability forecasts. *International Journal of Forecasting*, 27(2):208–223, 2008.

[22] José Luis Redondo García, Carlos Badenes, David Chaves Fraga, and Oscar Corcho. Final version of ontology learning and matching techniques with report. Technical report, 2016.

[23] Vasile Rus, Nobal Niraula, and Rajendra Banjade. Similarity Measures Based on Latent Dirichlet Allocation. In *Computational Linguistics and Intelligent Text Processing*, pages 459–470. Springer US, 2013.

[24] Pradip Kumar Sahu. *Research Methodology: A Guide for Researchers In Agricultural Science, Social Science and Other Related Fields*. Springer US, 2013.

[25] Ville a. Satopää, Jonathan Baron, Dean P. Foster, Barbara a. Mellers, Philip E. Tetlock, and Lyle H. Ungar. Combining multiple probability predictions using a simple logit model. *International Journal of Forecasting*, 30(2):344–356, 2014.

[26] Asha Singh and Balkeshwar Singh. Procedure of Research Methodology in Research Studies. *European International Journal of Science and Technology*, 3(9):79–85, 2014.

[27] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.