DR. INVENTOR SCIENTIFIC TEXT MINING LIBRARY: A SOFTWARE LIBRARY FOR DEEP ANALYSIS OF SCIENTIFIC DOCUMENTS

22/12/2017

UNIVERSITAT POMPEU FABRA
HORACIO SAGGION & FRANCESCO RONZANO

DR INVENTOR SCIENTIFIC TEXT MINING LIBRARY: A SOFTWARE LIBRARY FOR DEEP ANALYSIS OF SCIENTIFIC DOCUMENTS

Horacio Saggion & Francesco Ronzano

Universitat Pompeu Fabra

Barcelona, Spain

SUMMARY

This document describes the *Dr. Inventor Scientific Text Mining Library*, the collection of scientific text mining modules that enables the automatic extraction and aggregation of information from scientific publications. This document provides an architectural overview of the scientific text analysis modules integrated in the library, followed by a detailed description of each single module. Then we present the evaluation of the performance of four core scientific text analysis tasks implemented by four modules: the rhetorical classification of sentences, the classification of the purpose of citations, the generation of extractive summaries of scientific publications and the identification of causal relations.

1. DR. INVENTOR SCIENTIFIC TEXT MINING LIBRARY

The Dr. Inventor scientific text mining library (DRI for short) integrates in a single software platform a collection of scientific text analysis modules useful to automatically extract a varied range of structural, linguistic and semantic features from the textual content of scientific publications. The text analysis modules of the DRI library have been developed from scratch or adapted from existing text mining software and tailored to the scientific documents. Each module is responsible for the analysis of a particular aspect of the knowledge encoded in a scientific publication. Most of the times the processing results of a module are represented by means of textual annotations. Such annotations are in turn exploited by other modules to analyze further facets of scientific articles.

The DRI library constitutes a text mining tool to extract and model knowledge from scientific publications. The results of the scientific text mining performed by the DRI library enables a wide range of scientific literature analyses and data aggregations. Among its features, the DRI library supports the representation of excerpts of scientific papers by means of Subject-Verb-Object graphs, also referred to as Research Object Skeleton graphs (ROS).

In this Section we introduce the final release of the DRI library, distributed publicly as a self-contained Java library. In Section 2, we outline the general context of exploitation of the DRI library by discussing the problem of scientific information overload. Then we briefly review the main peculiarities that characterize the structure and content of scientific publications: we pose special focus on the facets of scientific articles that can be analyzed by the DRI library, thus motivating the relevance of such analyses to ease the exploration and study of scientific literature. After providing full details about the architecture of the DRI library in Section 3 (and through sections 3.1 to 3.12), we describe in Section 4 the possibilities for rich modeling of sentences for text mining. In Section 5 and Section 6 we provide details on how we respectively perform rhetorical sentence classification and citation purpose identification. Section 7 analyzes the

performance of the causality identification module. In Section 8, we show how to use the DRI framework in practice and in Section 9 we close this document with some conclusions.

2. *MINING PAPERS TO DEAL WITH SCIENTIFIC INFORMATION OVERLOAD*

Nowadays researchers can access on-line a huge amount of scientific literature that is rapidly growing: recent estimates reported that a new paper is published every 20 seconds [2]. PubMed[1], the reference publication index for life science and biomedical topics, currently includes more than 25 million papers: 1,370 new articles are added every day. The Cornell University Library arXiv initiative[2] provides access to over 1 million e-prints from various scientific domains.

In the meanwhile, the number of articles that are freely accessible on-line is considerably growing [3, 4]. More than 27% of the articles indexed by PubMed can be downloaded for free. The Directory of Open Access Journals[3], one of the most authoritative indexes of high quality, Open Access, peer-reviewed publications, lists more than 11,000 journals and 2.8 million papers. In 2011, 17% of the articles indexed by Scopus and ISI Web of Knowledge were freely available and this percentage is growing considerably [5]. Sometimes between 2017 and 2021, more than half of the global papers are expected to be published as Open Access articles [6]. Major publishing houses like Springer and Elsevier are currently increasing their portfolio of Open Access journals and initiatives. Moreover, well recognized conferences such as IJCAI, AAAI, Machine Learning, ACL just to name a few, are making their content freely available through dedicated archives even before the conference takes place.

In this scenario, researchers, as well as any other interested actor, are overwhelmed by an enormous and continuously growing number of articles to consider. The exploration of recent advances concerning specific topics, methods and techniques, peer reviewing, the writing and evaluation of research proposals and in general **any activity that requires a careful and comprehensive assessment of scientific literature has turned into an extremely complex, time-consuming task**.

Considering also the increasing amount of scientific information freely accessible on-line, **the availability of text mining tools able to extract, aggregate and turn scientific unstructured textual content into well organized and interconnected knowledge is fundamental**. In this context, the DRI library provides a coherent, self-contained scientific text analysis platform that enables the automated extraction of a wide range of structural and semantic information from scientific articles. To mine scientific publications, the DRI library has to properly deal with their many structural, linguistic and semantic peculiarities, thus substantially adapting and extending general purpose text mining tools and techniques. In respect to this, in the rest of the subsection, we provide an overview of the peculiar aspects that characterize scientific publications: we pose special focus on those facets of scientific publicationsaspects that can be analyzed by the DRI library, thus

[1] http://www.ncbi.nlm.nih.gov/pubmed

[2] http://arxiv.org/

[3] https://doaj.org/

motivating the relevance of such analyses for the exploration and study of scientific literature.

Even if the adoption of Web-friendly, textual formats and XML dialects like JATS[4] [7], Elsevier Schemas[5] and RASH[6] is rapidly spreading, *the majority of scientific papers is still available as PDF documents*, thus requiring proper tools to consistently extract their content [8, 9, 10, 11]. Scientific publications include *common structural elements* (title, authors, abstract, sections, figures, tables, citations, bibliography) that often require customized approaches to be properly characterized [12, 13, 14, 15]. Similarly, scientific articles are characterized by their *peculiar discursive structure* (background, challenge, outcome, future works) [16, 17]. Another characteristic aspect of papers is their *network of citations* that identifies un-typed links among pieces of work. Citation counts (e.g. h-index) currently constitute the core element that is used to evaluate the impact of a publication. Anyway, citation semantics has started to be exploited in several context including opinion mining [18, 19] and scientific text summarization [20, 21] and can represent the starting point to the creation of new, more accurate research evaluation approaches. Besides citations, the *interpretation of the semantics of the actual textual content of scientific papers* usually requires the availability of knowledge repositories with an adequate coverage of scientific concepts and relations that are often build by relying on and extending general domain knowledge resources like WordNet[7], DBPedia[8] or BabelNet[9].

Recently, several investigation and development efforts have been focused on the modelling and interlinking of scholarly publishing content by relying on Semantic Web standards and technologies [22, 23, 24]. This trend is usually referred to as *semantic publishing* [25]. Nowadays the bibliographic records of several publication repositories including DBLP[10], ACM[11] and IEEE[12] are already available as RDF Linked Data. Moreover, several projects are trying to model semantically more fine-grained information from scientific articles, including the venue of publication, the affiliation of authors, the funding bodies or relevant entities mentioned in their textual content. In this context, the Semantic Publishing Challenges [26], organized since 2014 as part of the Extended Semantic Web Conferences, represents an important venue to discuss and evaluate new approaches to the generation of scholarly publishing Linked Datasets.

3. *Architectural overview*

The DRI library is a scientific publication analysis platform that integrates several content analysis modules useful to characterize structural, linguistic and semantic aspects of articles. The DRI library is available to the scientific community as a self-contained Java library, thus making easier the experimentation of new approaches to analyze and

[4] http://jats.nlm.nih.gov/

[5] http://www.elsevier.com/author-schemas/elsevier-xml-dtds-and-transport-schemas

[6] https://rawgit.com/essepuntato/rash/master/documentation/index.html

[7] https://wordnet.princeton.edu/

[8] http://wiki.dbpedia.org/

[9] http://babelnet.org/

[10] http://dblp.l3s.de/d2r/

[11] http://acm.rkbexplorer.com/

[12] http://ieee.rkbexplorer.com/

aggregate content from collections of scientific texts. The latest version of the DRI library Java library, together with the related tutorials and documentation can be downloaded at the following URL:

http://taln.upf.edu/pages/dri.upf/

The DRI library Java library defines an *object-oriented data model* (i.e. set of classes) tailored to represent all the data extracted from scientific publications. By relying on this data model, users can trigger scientific text analyses as well as retrieve their results by a convenient *API*. The DRI library relies on the GATE Text Engineering Platform [1] to integrate its text mining modules as well as to internally manage and store text analysis results by means of textual annotations. Indeed, each module of the DRI library is modelled as a GATE Processing Resource[13].

In Figure 1 we provide an overview of the modules integrated in the DRI library that will be described in details in the next subsection.

FIGURE 1 - *ARCHITECTURE OF THE DR. INVENTOR TEXT MINING LIBRARY*

[13] https://gate.ac.uk/sale/tao/splitch7.html

3.1. *PDF-TO-TEXT CONVERTER*

As we can see from Figure 1, the DRI library is able to process papers in both PDF and JATS XML formats (it also deals with plain text in txt format). When we have to mine articles in PDF format, specific pre-processing actions are required to correctly extract structured textual content from PDF files. On the contrary these analyses are not needed when we process papers available as XML documents, since most of the XML formats for scientific publications (like JATS) already identify and provide direct access both to the text of the different excerpts of a paper as well as to other structural elements like in-line citations or bibliographic entries.

The *PDF to text converter* is the DRI library module responsible to extract structured textual content from the PDF file of a scientific publication. After reviewing and testing several PDF-to-text conversion approaches both generic and tailored to scientific publications, we decided to rely on GROBID[14]: it is a machine learning tool that supports the extraction and structural characterization of textual content from scientific articles that are available as PDF documents. GROBID is implemented in Java: PDF files are converted to text by means of the pdf2xml tool[15]. Then, a chain of sequence taggers are exploited to spot a complex set of structural elements inside publications including title, sections, abstract, footnotes, etc. (see Figure 2). The sequence taggers exploited by GROBID are Conditional Random Fields models managed by relying on the C++ libraries CRF++[16] and Wapiti[17] and trained over a corpus of manually annotated scientific papers. GROBID is an open-source project[18] exploited (after extensive customizations) by several scientific publishing companies including ResearchGate.

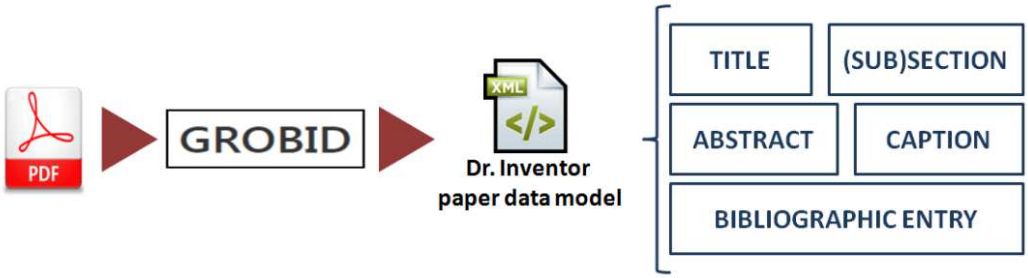

FIGURE 2 - PDF-TO-TEXT CONVERSION BY MEANS OF PDFX WEB SERVICE

3.2. *IN-LINE CITATION SPOTTER*

[14] https://grobid.readthedocs.io/en/latest/

[15] https://github.com/kermitt2/pdf2xml

[16] https://taku910.github.io/crfpp/

[17] https://wapiti.limsi.fr/

[18] https://github.com/kermitt2/grobid

Once converted to text, the content of a PDF paper is analyzed by means of *the In-line citation spotter* that performs the following tasks (see Figure 3).

- **task A**: identification of inline citation spans and markers (see Figure 5) in the textual content of the paper by means of a set of JAPE rules[19] [27], tailored to match widespread inline citations styles;
- **task B**: identification of the bibliographic entries, usually listed at the end of the paper. The validity of the bibliographic entries identified by PDFX is verified;
- **task C**: linking of each inline citation marker to the referenced bibliographic entry by means of a set of heuristics. For instance, to determine the bibliographic entry referenced by an inline citation marker that contains the first author surname and the publication year, we select the bibliographic entry with the highest number of tokens in common with the same marker. While, if the inline citation marker references a bibliographic entry by a number or a short string, we link this marker to the bibliographic entry that has such number or short string in its first 15 characters;
- **task D**: identification of the syntactic/non-syntactic role of each inline citation marker, in order to properly support the dependency parsing of the sentence in which the citation marker occurs (see following text analysis components). The first inline citation marker of Figure 5 has a syntactic role in the sentence (subject), while the second one has no syntactic role. To verify the syntactic role of an inline citation span we exploit the approaches described in [20] and [28].

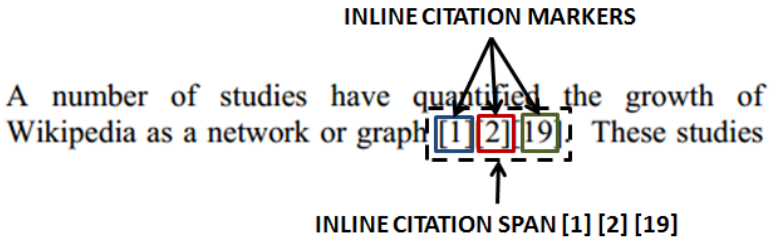

FIGURE 3 - INLINE CITATION MARKERS AND SPANS

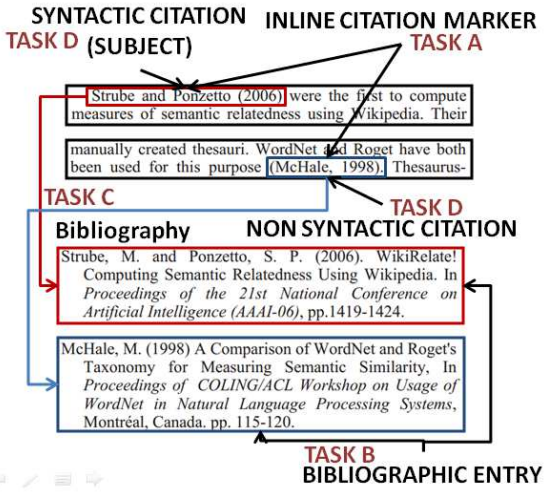

FIGURE 4 - PROCESSING STEPS OF THE IN-LINE CITATION SPOTTER

[19] https://gate.ac.uk/sale/tao/splitch8.html

3.3. *SENTENCE SPLITTER*

The *Sentence splitter module* spots sentences of scientific publications by identifying their boundaries. To this purpose, we customized the rule-based sentence splitter integrated in ANNIE[20], the information extraction system bundled in GATE. We analyzed the sentence split errors performed on the set of 40 Computer Graphics papers of the DRI Corpus (occurring with expressions like: i.e., et al., Fig., Tab.) and modified the sentence splitting rules of ANNIE in order to correctly deal with these situations.

3.4. *WEB-BASED REFERENCE PARSER*

The Web –based reference parser invokes Web Services to parse or retrieve descriptive metadata of the bibliographic entries of a scientific paper, including its title, the names of the authors, the year of publication, the venue or journal of publication, etc. (see Figure 5)

The following Web Services are queried so as to analyze the content of bibliographic entries:

- **FreeCite**[21]: this on-line tool analyzes citations by relying on a conditional random field sequence tagger trained on the CORA dataset, made of 1,838 manually tagged bibliographic entries[22];
- **Bibsonomy**[23]: its Web API enables the retrieval of the BibTeX metadata of a publication from the Bibsonomy database by providing its title as the query string.

FIGURE 5 - PARSING OF A BIBLIOGRAPHY ENTRY

We merge the results retrieved by querying the REST endpoints of these two Web services, trying to determine for each bibliographic entry the title of the paper, the year of publication, the list of authors, and the venue or journal of publication. We give precedence to Bibsonomy results over Freecite output, when the outputs of their responses disagree; this is due to the greater accuracy of the bibliographic entry metadata search performed by Bibsonomy (empirically studied by analyzing the outcome of several

[20] https://gate.ac.uk/sale/tao/splitch6.html\#chap:annie

[21] http://freecite.library.brown.edu/welcome

[22] https://hpi.de/naumann/projects/repeatability/datasets/cora-dataset.html

[23] http://www.bibsonomy.org/help/doc/api.html

papers). We have to observer tha Bibsonomy tries to find in its bibliographic database, the record that best matches the content of a bibliographic entry. On the contrary, FreeCite is based on a machine learning approach: by training a conditional random field sequence tagger on 1,838 manually tagged bibliographic entries, FreeCite is able to automatically spot, inside the content of a bibliographic entry, which words belong to the title, to the author, which is the year of publication and so on. The results of FreeCite are considered only if the bibliographic entry to analyze doesn't match any record of the Bibsonomy bibliographic database.

3.5. *CITATION-AWARE DEPENDENCY PARSER*

The *Citation-aware dependency parser* executes the dependency parsing of the sentences of a paper by properly dealing with sentences that include inline citations. To this purpose, we rely on **MaltParser**[24] [29], a data-driven parser-generator exploited to determine the syntactic structure of the sentences of a paper.

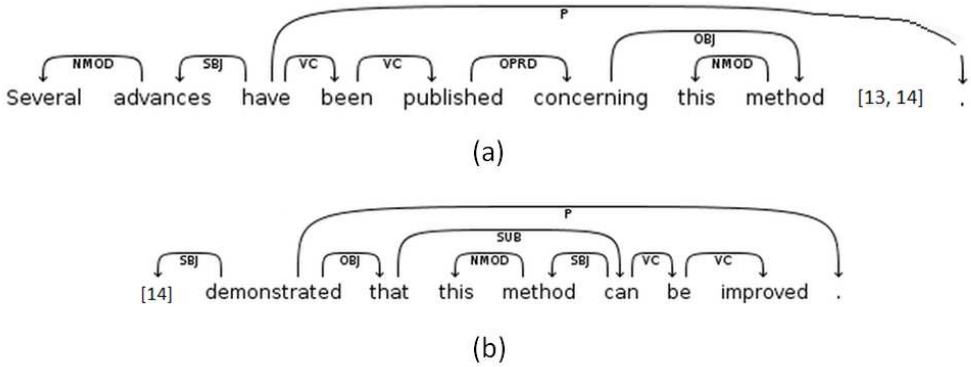

FIGURE 6 – SYNTACTIC PARSING OF SENTENCES IN SCIENTIFIC ARTICLES

We modified the parser to correctly deal with inline citation spans when building the dependency tree of a sentence. We exclude inline citations from the textual content to parse if they have no syntactic role in the sentence to analyze. As commented, the In-line citation spotter is responsible of identifying the syntactic role of in-line citations. Figure 6 shows two examples of dependency tree of sentences including an in-line citation. In the first sentence, the in-line citation ***"[13, 14]"*** has no syntactic role and is thus ignored by the parser. On the contrary, the in-line citation of the second sentence ***"[14]"*** represents the subject of the same sentence and thus actively contributes to the structure of the dependency tree.

3.6. *RHETORICAL ANNOTATOR*

The *Rhetorical annotator* automatically classifies each sentence of a paper by associating a specific rhetorical category among: ***Approach***, ***Challenge***, ***Background***, ***Outcomes*** and ***Future Work*** (see Figure 6). Meta-discourse sentences (like the description of the organization of the paper, the acknowledgments, etc.) are usually assigned as ***Unspecified***. This module relies on as support vector machine classifier, **LIBSVM**[25] trained on the

[24] http://www.maltparser.org/

[25] **https://www.csie.ntu.edu.tw/~cjlin/libsvm/**

papers of the Dr. Inventor Multi-layered Corpus of Scientific Publications [30, 31]. In Section 5, we present the features the library extracts for implementing a SVM classifier.

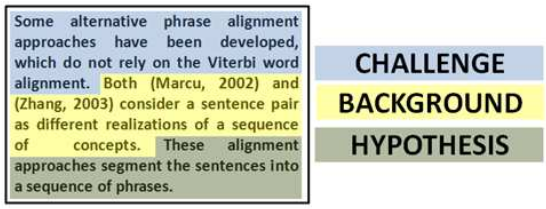

FIGURE 6 - THE RHETORICAL ANNOTATION OF SENTENCES

3.7. *CITATION PURPOSE ANNOTATOR*

The *Citation purpose annotator* is responsible for the automated classification of the purpose of each sentence including one or more in-line citations (citing sentence). In particular, we classify each citing sentence as belonging to one of these citation purpose categories: ***Criticism***, ***Comparison***, ***Use***, ***Substantiation***, ***Basis*** and ***Neutral*** (see Figure 7). Similarly to the Rhetorical annotator module, this module also relies on the **LIBSVM** library to identify the purpose of citations and has been trained on the papers of the Dr. Inventor Multi-layered Corpus of Scientific Publications [30, 31]. When annotating the citation purpose of a sentence containing one or more in-line citations we make the following simplifications:

- the whole citing sentence expresses a single citation purpose that is the same irrespective of the number of in-line citations included in that sentence;
- we consider the content of a citing sentence sufficient to characterize the citation purpose of the same citing sentence, without contemplating any surrounding sentence.

Section 6 presents the classification approach.

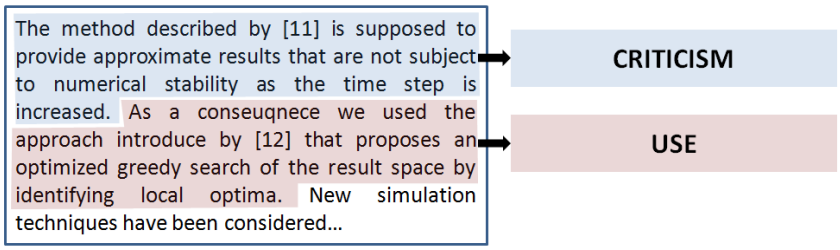

FIGURE 7 - THE CITATION PURPOSE ANNOTATION OF CITING SENTENCES

3.8. *RDF GENERATOR*

The RDF generator is responsible for assembling a semantic representation of the content of a scientific publication as an RDF Linked Dataset, in accordance with the semantic publishing principles [25]. When we model the data of a paper by means of a set of RDF triples, we extend and enrich the basic RDF data modeling approach of scientific papers we adopted in the context of our participation to the Semantic Publishing Challenge 2015 [34]. Our RDF data modeling choices have been driven by the necessity to represent the varied set of information that can be mined from a publication by relying on the modules

of the DRI library. As a consequence, besides the representation of articles' metadata and bibliographic entries, we represent by means of RDF triples the structure of a paper, by identifying its abstract, sections and sentence. Each sentence is characterized by its rhetorical category, identified by the Rhetorical annotator. Moreover we link each bibliographic entry of a paper to all the sentences that include the related in-line citations.

To represent the content of a scientific publication we rely on the core RDF data modeling approaches, patterns and ontologies accessible in the Semantic Publishing and Referencing (SPAR) Portal[26] [35]. The SPAR Portal defines and documents a complete and consistent set of 12 ontologies tailored to model several aspects of scientific publishing, including articles' metadata, authors, bibliography, citations, publication workflows, etc. From the classes and the properties modelled by the SPAR ontologies, we reused and derived - in the dri namespace - new sub-classes and sub-properties. As a consequence, we included the related T-BOX axioms in the RDF Datasets we generate. To enable the conversion of the information extracted from a paper into RDF, a namespace under our control has to be specified. In this way we can generate the URIs needed to unambiguously reference the article and its components (authors, sections, sentences, bibliographic entries, etc.). This information is usually provided by DRI library users when the RDF generation process is invoked.

Figure 8 contains: (a) authors and internal structure of the paper including sections and sentences with rhetorical classes, (b) list of bibliographic entries together with the pointer to the sentences in which the biblio reference occurs, and (c) descriptive data of papers and biblio entries.

The ontology prefixes used are: DOCO Document Components Ontology, FABIO FRBR-aligned Bibliographic Ontology, C4O Ccitation Counting and Ccontext Characterization Ontology, PRO Publishing Roles Ontology, BIRO Bibliographic Reference Ontology, SWRC Semantic Web for Research Communities ontology, PRISM PRISM metadata ontology, FOAF Oriend Of A Friend ontology, PO Pattern Ontology, CO Collections Ontology, DC and DCTERMS Dublin Core ontology. The prefix dri identifies the classes and properties of Dr. Inventor ontology.

Figure 8 (a) provides a detailed representation of our RDF data modelling approach. In particular, in (a) we schematize how we represent the structure of the content of a paper as RDF triples. Two URIs are generated to reference the abstract and the body of the paper (respectively the *FrontMatter_URI* and the *BodyMatter_URI*). Both the abstract and the body may contain a list of sections (*IntroSection_URI* and *MethodSection_URI*). Each section is assigned an URI and related to an instance of the doco:SectionTitle class that represents its title. The abstract, body and sections of the paper can contain one or more sentences, each one identified by an URI (*Sentence1_URI*, ..., *SentenceN_URI*). In the lower part of (a) we represent the association of the sentences of the paper to their scientific discourse rhetorical category. This is achieved by representing the corresponding *sentence_URI* as an instance of one of the following classes: dri:Approach, dri:Challenge, dri:Background, dri:Outcomes and dri:FutureWork, instantiated in the Dr. Inventor Scientific Discourse Ontology [36].

[26] http://www.sparontologies.net/

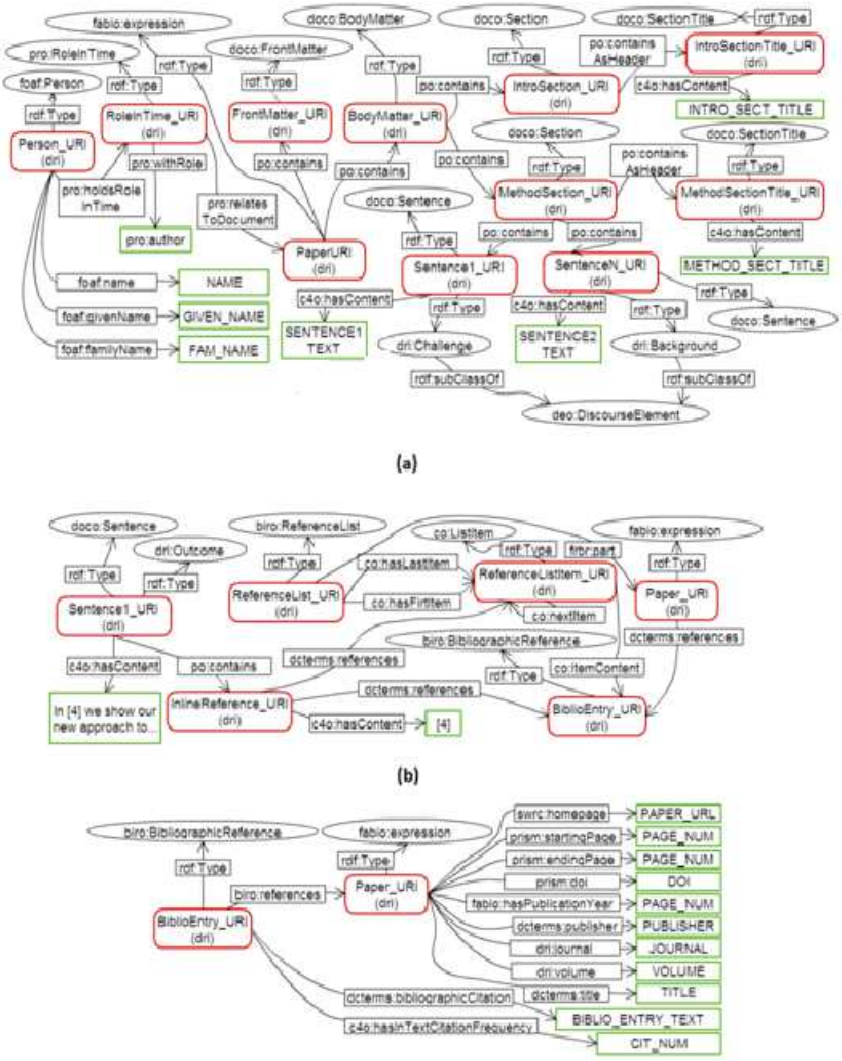

FIGURE 8- RDF DATA MODEL OF SCIENTIFIC ARTICLE

.

3.9. *CAUSALITY SPOTTER*

The *Causality spotter* is responsible to identify causal relations inside the text of a scientific publication. The recognized causal relations can be exploited to enrich the expressiveness of triples extracted from the analyzed documents.

Each causal relation is a natural language statement that connects a text excerpt that represents the cause to another text excerpt that represents the related effect. A set of JAPE rules[27] [27] has been developed to identify the presence of a causal relation inside the textual content of a paper. Each JAPE rule looks for a specific pattern commonly used to express a causal relation by analyzing distinctive lexical and linguistic features of the

[27] https://gate.ac.uk/sale/tao/splitch8.html

textual content of a paper together with its dependency relations spotted by the Citation-aware dependency parser. The JAPE rules implemented by the *Causality spotter* rely also on custom lists of cue phrases and expressions that usually point out the presence of a causal relation.

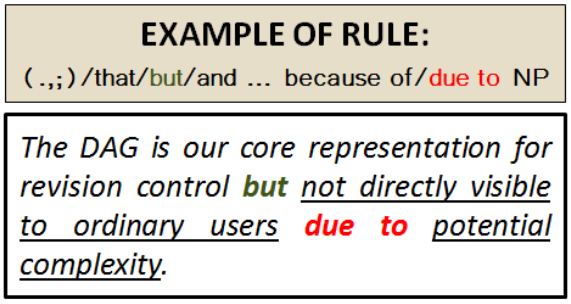

FIGURE 9 - EXAMPLE OF JAPE RULE USEFUL TO SPOT CAUSAL RELATIONS

We identified and iteratively refined the *Causality spotte*r JAPE rules by analyzing the textual content of the 40 papers included in the Dr. Inventor Multi-layered Corpus. This process resulted in the creation of a collection of 84 JAPE rules grouped in 39 grammars. In Figure 9 we provide an example of a causality spotting JAPE rule. The upper part of Figure 9 shows the linguistic pattern that is matched by the rule: to trigger this rule the textual content analyzed should contain a full stop (.), a colon (:), a semicolon (;) or a word among *that*, *but* or *and*, followed by a set of words, in turns followed by *because of* / *due to* and a Noun Phrase (NP). In the lower part of Figure 11 we can see an example of match of the causality rule just described where we have identified the cause 'potential complexity' and the effect 'not directly visible to users'.

Both the cause and relation text excerpt identified by the *Causality spotter* module can represent a Noun Phrase or a more complex expression covering also a whole sentence.

The JAPE rules implemented by the *Causality spotter* module contemplate also the identification of ***Cross-sentence causal relations*** in which the effect is expressed in a sentence different from the one that contains the cause or vice versa. For instance, we can consider the following sentences:

> *The proposed algorithm can be distributed across different machines. This is due to the parallel execution optimization we performed.*

In this case the cause is 'the parallel execution optimization' and the related effect is explained in the previous sentence (and referenced by the word 'this'). As a consequence, the *Causality spotter* will mark as the effect the text 'The proposed algorithm can be distributed across different machines'.

3.10. *COREFERENCE RESOLUTOR AND GRAPH BUILDER*

The *Coreference resolutor and graph builder* is responsible for the representation of textual excerpts of scientific publications by means of subject-verb-object (SVO) graphs. The SVO graph of a textual excerpt (i.e. part of the content of a paper) is a graph in which the nodes represent nominal and verbal expressions and the arcs identify relations among them: in particular, three kinds of relations are contemplated:

- **Subject**: connecting a nominal node with the verbal node that represent the verbal expression the nominal node is subject of;
- **Object**: connecting a nominal node with the verbal node that represent the verbal expression the nominal node is object of;
- **Cause**: connecting a nominal or verbal node that represents the cause to a nominal or verbal node that represents the related effect.

The creation of the graph that represents the content of a paper is based on:

- the dependency graph built from each sentence of the paper by means of the *Citation-aware dependency parser* module;
- the causality relations identified by the *Causality spotter* module.

In order to increase the connectedness of graphs, we have implemented and integrated in DRI library a coreference resolutor by relying on the deterministic coreference resolution approach proposed by the Stanford Coreference Resolution System[28] [37]. The final aim of the coreference resolutor is to identify groups of nominal expressions that refer to the same entity, called co-referents. All the co-referent nominal nodes of a graph identified thanks to the coreference resolutor can be merged into a single node, thus increasing the graph connectedness. The possibility to generate bigger connected components of the graph by avoiding the duplication of nominal nodes that refer to the same entity enables more consistent and complete analogical comparisons among graphs of different papers.

The coreference resolutor implemented in the DRI library processes the content of a scientific publication by means of the following sequence of two steps (see Figure 10) in order to identify groups of nominal expressions that refer to the same entity (coreferents) thus creating a coreference chain.

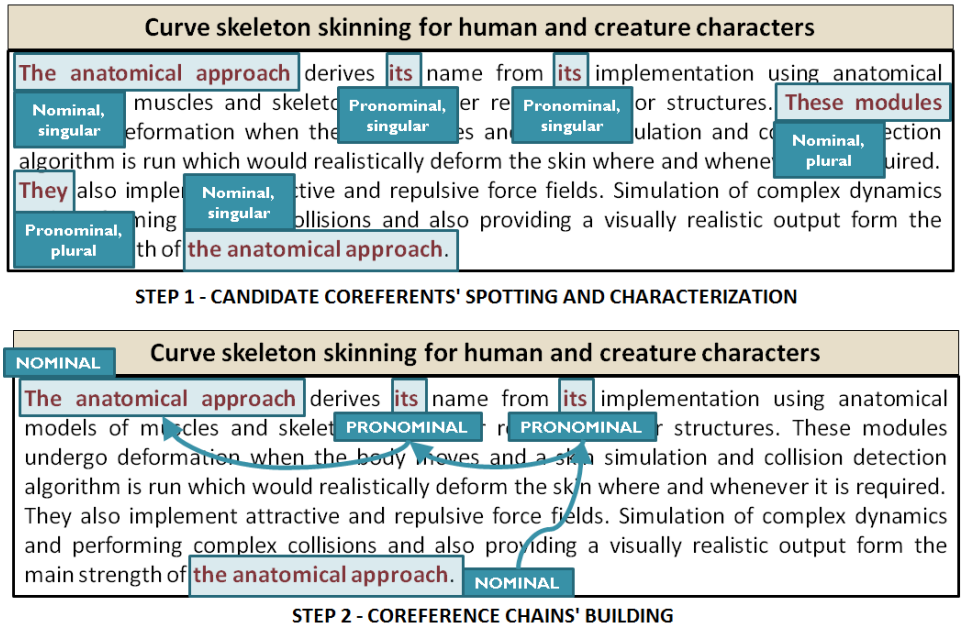

FIGURE 10 - THE TWO TEXT ANALYSIS STEPS OF THE COREFERENCE RESOLUTOR

[28] http://nlp.stanford.edu/software/dcoref.shtml

In a first step (STEP 1 in Figure 10), candidate nominal and pronominal expressions (i.e. candidate coreferents) are spotted and characterized by means of a set of linguistic and semantic features (gender, number, occurrence in predefined lists of words, etc.). Then, a second processing step (STEP 2 in Figure 10) applies a set of heuristics and rules in order to group together the candidate coreferents that refer to the same entity thus creating coreference chains. We consider corefrence chain creation rules that take into account both nominal and pronominal matching of candidate coreferents. In the STEP 2 of Figure 10, we can notice that the coreference chain shown is composed by four coreferents, two nominal one ('the anatomical approach') and two pronominal ones ('its').

In Figure 11 we provide an example of a graph generated by means of the *Coreference resolutor and graph builder* module, stressing the contribution of different components of the DRI library in modelling the same graph. The box on the left side of Figure 11 shows a textual excerpt taken from a scientific article. On the right side of Figure 11, the corresponding ROS graph is shown.

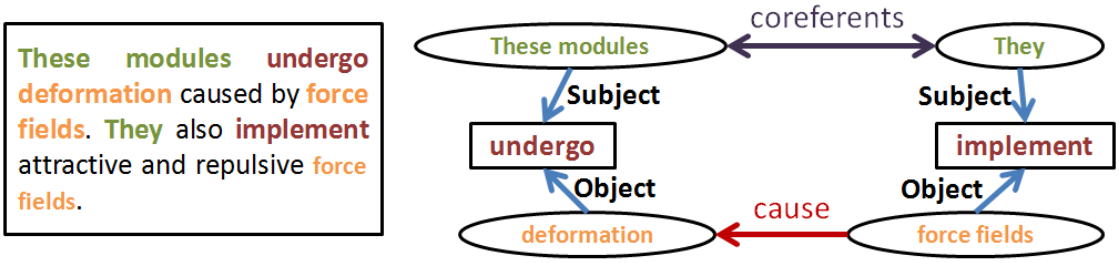

FIGURE 11 - EXAMPLE OF GRAPH GENERATED BY MEANS OF THE COREFERENCE RESOLUTOR AND GRAPH BUILDER MODULE

The text excerpt shown in Figure 11 is composed by two sentences. Thanks to the Citation-aware dependency parser module we can populate the graph with the Subject-Verb-Object relations extracted from both sentences: 'These modules' is subject of the verb 'undergo' that in turns has as direct object 'deformation' and 'They' is the subject of the verb 'implement' that has as direct object 'force fields'. By relying on the Causality spotter module, we can add to the graph the causal relation between the nominal node 'force fields' (cause) and the nominal node 'deformation' (effect). Moreover, thanks to the coreference resolutor we are able to identify that the pronoun 'They' refers to the nominal node 'These modules' (i.e. they are coreferents): as a consequence we are able to merge both nodes into a single one in the final graph.

3.11. *EXTRACTIVE SUMMARIZER*

The *Extractive summarizer* implements extractive summarization approaches both generic and tailored to scientific publications by relying on the content summarization facilities provided by SUMMA toolkit[29] [38]. Each summarization approach implements a specific strategy to generate a summary of a paper by selecting a subset of relevant sentences (see Figure 12).

[29] http://www.taln.upf.edu/pages/summa.upf/

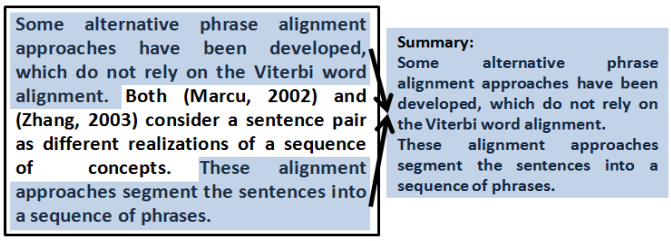

FIGURE 12 - EXTRACTIVE SUMMARIZATION

The current version of the *Extractive summarizer* implements basic extractive summarization approaches such as the following:

- **TITLE-BASED SUMMARIZATION**: we rank the sentences of each publication with respect to their **tf*idf** vector similarity with the title of the same publication and choose the most similar sentences to generate the summary of that article;
- **CENTROID-BASED SUMMARIZATION**: we compute the centroid of all the **tf*idf** vectors, each one associated to a sentence of the paper to summarize. The sentences that are characterized by a tf*idf vector most similar to the centroid vector are chosen to be included in the summary of that article;
- **LEXRANK SUMMARIZATION**: based on the LexRank extractive summarization approach [8], a graph-based method for computing relative importance of textual units. By considering non-abstract sentences as textual units, we build a graph where each sentence of a paper is a node. Pairs of sentence nodes are connected by an ark with weight equal to the cosine similarity of the td*idf vectors of the sentences, if the value of the cosine similarity is greater than 0.1. The LextRank algorithm is applied to such graph: by relying on random walks and eigenvector centrality the most relevant sentence nodes of the graph are selected and the corresponding sentences are then grouped to generate a summary of the paper.

Further methods can be easily integrated in the library.

3.12. *TRANSVERSAL FEATURES OF THE JAVA LIBRARY*

In this section we provide a brief overview of some transversal feature of the Java library that implements the DRI library.

- **VERSIONING AND DATA PERSISTENCE**: the DRI library gives the possibility to programmatically access the processing results of a scientific publication by means of the methods exposed by the edu.upf.taln.dri.lib.model.Document interface. The processing results of a scientific publication can be persisted in the form of an XML file so as to be reloaded in memory when needed again. The DRI library Java library transparently manages the persistence of processed papers as XML files across versions. In particular, let suppose that a paper is processed by a version A of the library and stored as an XML file. If the processing results of that paper are loaded from the XML file by using a newer version of the library, the DRI library versioning system will re-execute only of the subset of the text mining modules that have been changed in the new version of the library with respect to version A. In this way the reprocessing of data among papers persisted across different versions of the library is minimized.
- **LAZY-PROCESSING**: all the analyses performed over the content of a paper by the modules of the DRI library are triggered only when its results are required by the user for the first time (for instance by invoking one of the methods exposed by the

edu.upf.taln.dri.lib.model.Document interface). In this way, no data processing that is not actually required is carried out. The DRI library java library provides also a method of the edu.upf.taln.dri.lib.model.Document interface – preprocess() – that triggers all the analysis available of a paper and stores in memory the results that are ready for their consumption by the user.

- **PARALLEL EXECUTION OF MODULES**: each module of the DRI library Java library cannot process documents in parallel. To manage this constraint, each module of the DRI library has an associated FIFO waiting queue. When a document A needs to be analyzed by means of a module that is analyzing a document B, the document A is added to FIFO waiting queue of the module and processed as soon as the analysis of B ends. As a consequence, all the modules of the DRI library can process different documents in parallel, but each single module can not process two or more documents in parallel.

4. *EXTRACTING FEATURES FOR CLASSIFICATION WITH THE LIBRARY*

In this Section we present the evaluation of the core modules of DRI library, presented in Section 3: the *Rhetorical annotato*r, the *Citation purpose annotator*, and the *Causality spotter*. Since the evaluation of the first three modules (over the four just listed) relies on machine learning approaches trained on the manual annotation of the Dr. Inventor Multi-layered Corpus, in Section 4.1 we provide an overview of the structure of this Corpus.

4.1. *GOLD STANDARD DATASET: THE DR. INVENTOR MULTI-LAYERED CORPUS*

The Dr. Inventor Multi-layered Corpus [30, 31] includes 40 Computer Graphics papers, selected and collaboratively annotated by domain experts (see Figure 13). The papers of the DRI Corpus are divided into four groups of 10 articles. The papers of each group deal with a specific field of Computer Graphics: Cloth Simulation, Fluid Simulation, Motion and Skinning. The papers of the Corpus have been annotated with respect to several aspects of scientific literature (each one referred to as an annotation layer) by exploiting two tools: the GATE text engineering desktop platform [1] and Annote[30], a Web-based annotation framework we developed to support complex annotation tasks, specific to our Corpus.

In the remaining part of this section we provide an overview of the different aspects of scientific information that have been manually annotated in the papers of the Corpus (see Figure 13): the S*cientific discourse layer*, the *Citation purpose layer* and the *Scientific summarization layer*. The whole Corpus can be downloaded at: http://sempub.taln.upf.edu/dricorpus.

[30] http://penggalian.org/annote/

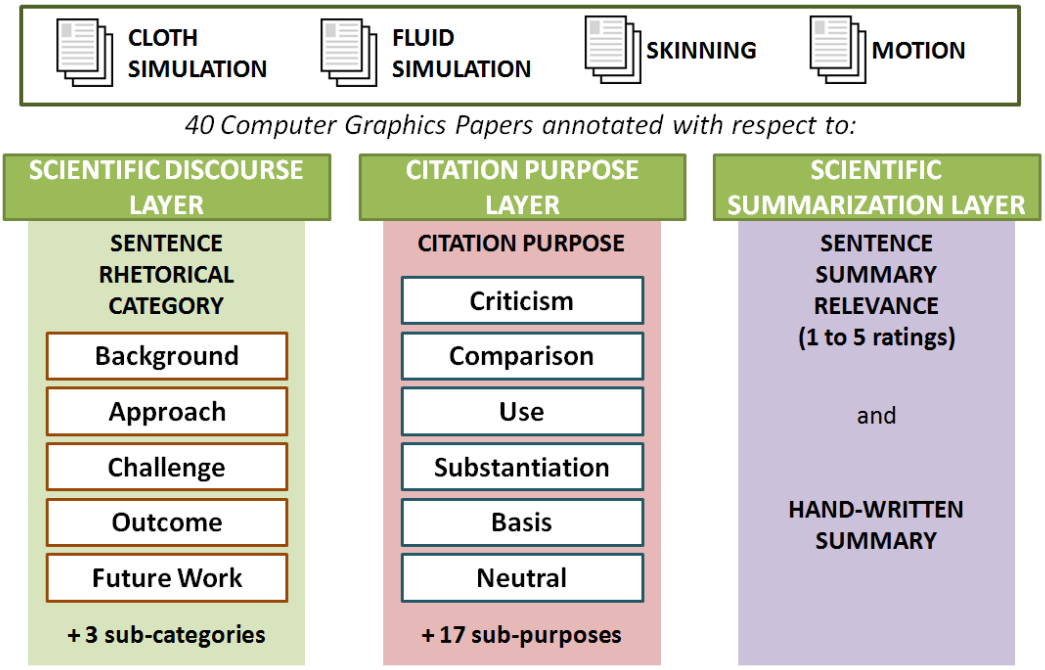

FIGURE 13 - STRUCTURE OF THE DR. INVENTOR MULTI-LAYERED CORPUS

4.2. *THE SCIENTIFIC DISCOURSE LAYER*

The identification of the discursive structure of scientific publications enables new, most effective patterns to aggregate and search for relevant content. For instance, by identifying the parts of a collection of papers that deal with research contributions, we could restrict our searches to these excerpts thus easily and precisely selecting all the articles that provide new results with respect to a specific topic. In order to develop automated approaches to explicitly characterize the discursive structure of a paper, the articles of Dr. Inventor Multi-layered Corpus have been manually annotate by associating to each sentence its Rhetorical Category.

After an extensive review of the rhetorical annotation schemas proposed in literature to characterize the content of scientific publications we decided to adapt both Liakata's CoreSc schema [16] and Teufel's Argumentative Zoning approach [17], thus defining a rhetorical annotation schema composed of the five top-level categories shown in Figure 15, complemented by three sub-categories. In particular, we defined two sub-categories for the sentences tagged as *Challenge*. These two sub-categories are useful to identify if the challenging aspect described is related to a *Hypothesis* or to one of the *Goals* of the article. We also defined *Contribution* as a sub-category of sentences identified as *Outcome*. In this way we can specify it an *Outcome* sentence describes or not to a contribution made by the authors of the paper. Annotators had also the options to assign the class *Unspecified* to the sentences that didn't fit in any of the available categories (for instance in case of meta-discourse or acknowledgements) and the class *Sentence* to the portion of texts that were incorrectly identified due to errors in the Sentence splitter. Three annotators classified a total of 10,789 sentences with an inter-annotator agreement (Cohen's k), averaged among all pairs of annotators equal to 0.6567 if we consider the 5 top categories and the 3 sub-categories. The inter-annotator agreement rises up to 0.6823 if we consider only the 5 top-categories. The distribution of sentences across rhetorical categories is shown in Table 1.

Rhetorical category	Number of sentences	Percentage of sentences (%)
Approach	5,038	46.70
Background	1,760	16.32
Challenge	351	3.25
Challenge_Goal	91	0.84
Challenge_Hypotesis	7	0.06
Future Work	136	1.26
Outcome	1,175	10.89
Outcome_Contribution	219	2.03
Unspecified	759	7.04
Sentence	1,253	11.61
Total:	10,789	

TABLE 1 - DISTRIBUTION OF ANNOTATED SENTENCES ACROSS RHETORICAL CATEGORIES

4.3. *THE CITATION PURPOSE LAYER*

The network of citations across papers constitutes one of the most distinctive traits of scientific articles. Each citation represents a strong explicit connection, provided by the authors of the citing paper, to other somehow related, relevant works. The count of the citations that a paper receives together with the identification of specific parameters and measures of the related citation network still constitute the basis of the most widespread metrics used to evaluate the scientific production of papers, journals and researchers (i.e. h-index, g-index, impact factor, etc.), even if alternative research evaluation metrics are gaining increasing relevance [40].

To go beyond the mere count, several studies have explored the possibility to characterize the semantics of citations by taking into account facets related to polarity and purpose. There are several reasons that can motivate the choice to cite a specific paper: we can cite to reference related works, to point out research outcomes we want to compare with, to criticize, etc. Several schemas, with different levels of granularity, have been proposed to characterize the purpose of citations [19, 40]. Relying on the 6 top-level citation purposes identified by [19] (shown in Figure 15), we annotated the sentences of the papers of the Dr. Inventor Multi-layered Corpus. In particular, for each group of in-line citations spotted by the DRI library (i.e. *[1]* or *[1-3]* or *[Rossi et al. 2013, Karl et al. 2015]*), we identified one of the 6 top-level citation purposes to characterize the sentence in which the group of in-line citations occurs and, if appropriate, each surrounding sentence inside the same section of the paper, considering a [-3,3] window (three sentences before and three sentences after the one under analysis). The sentence that contains a group of in-line citations together with the surrounding sentences that are useful to characterize the purpose of that citation group are referred to as citation context. A total of 2,356

annotated citation context sentences are included in the DRI Corpus. The distribution of sentences across different citation purposes is shown in Table 2.

Citation purpose	Number of sentences	Percentage of sentences (%)
CRITICISM	599	25.4
USE	300	12.7
SUBSTANTIATION	62	2.6
COMPARISON	211	9
NEUTRAL	983	41.7
BASIS	201	8.6
Total:	2,356	

TABLE 2 - DISTRIBUTION OF ANNOTATED CITATION CONTEXT SENTENCES ACROSS CITATION PURPOSES

4.4. *THE SCIENTIFIC SUMMARIZATION LAYER*

Effective approaches to identify the most relevant content both in a publication as well as in a collection of articles constitute an essential device to perform any comprehensive review or screening of scientific literature. During the last few years, several approaches to scientific summarization have been proposed [20, 42, 43]. Most of them extend general-purpose summarization methodologies by taking advantage of information facets that are characteristic of scientific publications. One valuable source of information useful to improve the quality of scientific summarization is represented by the sentences of papers in which the article to summarize is cited as explored by the Computational Linguistic Scientific Summarization Shared Tasks (last edition organized in the context of the Joint Conference of Digital Libraries 2016) [41]. Moreover, the possibility to consider the discursive structure (background, approach, future work, etc.) of the different excerpts of a paper to summarize provides additional relevant information to generate summaries that include content better balanced across the different sections of a paper.

In order to provide a useful dataset to evaluate scientific document summarization approaches, the annotators of the Dr. Multi-layered Corpus were asked to evaluate the relevance of the different content of a paper with respect to their inclusion in the summary of the same publication. In particular, considering the body (thus excluding the abstract) of each paper of the Corpus, each annotator associated to each sentence a summary relevance value. Such value is an integer number in the interval [1,5]. A higher summary relevance value associated to a sentence signifies a higher importance of that sentence with respect to its inclusion in a summary of the paper where it occurs. A total of 10,136 sentences of papers were rated. As expected, more than half of these sentences (66%) were classified as 'Totally irrelevant for a summary' (summary relevance score equal to 1). Only 8 sentences over 100 were given the maximum summary relevance score, 5, and thus were judged as 'Very relevant for a summary'. Besides the scoring of sentences, for each paper of the DRI Corpus, each annotator was asked to write a handwritten summary of maximum 250 words in length. As a consequence, the DRI Corpus includes three hand-

written summaries for each paper, useful as Gold Standard references to evaluate the outcome of automated summarization approaches.

5. *RHETORICAL SENTENCE CLASSIFICATION APPROACH*

The automated characterization of the discursive structure of scientific publications is one of the key challenges faced by the scientific text mining community. In this Section we explain how the library can be used to implement rhetorical sentence classification by utilizing supervised machine learning.

5.1. *RELATED WORK ON RHETORICAL CLASSIFICATION*

Several approaches have been proposed in literature to automatically determine the discursive structure of research papers. Most of them rely on supervised or semi-supervised classification or sequence labeling methods trained over a corpus of papers in which each sentence is manually or automatically assigned to a scientific discourse category. The main differences among the various approaches are:

- **Corpus**: the domain and size of the annotated corpus of scientific discourse;
- **Annotation schema and procedure**: the types of scientific discourse categories considered to annotate the corpus and the way the annotation process was conducted;
- **Features**: the features that were used to describe each textual excerpt (usually sentences) to be assigned to a scientific discourse category;
- **Algorithms**: the machine learning approaches exploited to determine the scientific discourse category.

[44] evaluated the performance of the Naïve Bayes classifier to determine the rhetorical category of sentences. To this purpose, they exploited the Argumentative Zoning Corpus[31], a set of 80 computational linguistics papers that were manually annotated by assigning to each sentence a rhetorical status from a schema (AZ) including 7 different rhetorical categories. In their experiments, they represented each sentence by means of a set of positional, structural and syntactic features including the presence of citations and the presence of tailored set of action verbs, agentive and formulaic expressions. [64] exploited the Teufel's annotations schema to annotate a corpus of 25 Portuguese articles and evaluated the performance of the Naïve Bayes classifier over this corpus.

[65] performed some sentence labeling experiments by relying on a corpus of 1,000 abstracts of biomedical papers annotated with respect to Teufel's AZ schema. They compared the performance of a Support Vector Machine classifier with a Conditional Random Field sequence labeling approach. They selected the best performing positional and syntactic features from previous works to represent the sentences. In addition, they also evaluated the performance of four weakly-supervised sentence tagging approaches in order to mitigate the availability of small amounts of labelled data in the training corpus. [66] exploited a corpus of 50 biomedical articles (8,171 sentences) annotated with respect to Teufel's AZ schema and evaluated the performance of both fully supervised sentence classification approaches (namely Support Vector Machine) and active learning approaches. In the latter case, they compared three different strategies of unlabelled

[31] http://www.cl.cam.ac.uk/~sht25/AZ_corpus.html

sample selection to iteratively increment the size of the training set. The features set they used to represent each sentence is similar to the one exploited by [65].

[67] carried out its experiments by relying on Teufel's Argumentative Zoning Corpus as well as the Astronomy Bootstrapped Corpus, a collection of 209 abstracts from the NASA Astronomical Data System Archive. They compared a Naïve Bayes classifier and a sequence-aware labeling approach. They described each sentence to classify by a subset of the features used by [44] extended with unigrams, bigrams and features spotting the presence of Named Entities.

[68] compared two classifiers, Naïve Bayes and Support Vector Machine, with respect to the classification of the rhetorical class of the sentences of the ZAISA-1 corpus, including 40 full text molecular biology articles (3,637 sentences). The sentence annotation schema adopted was based on [69]. The features to describe each sentence to classify included, besides lexical and syntactic ones, also morphological information about the main verb and the location of the sentence inside the paper. They evaluated also the relevance, with respect to the classification task, of the rhetorical class of the surrounding sentences.

[70] used a sequence labelling approach (Conditional Random Fields) to classify the sentences of the abstract of scientific publications into one over 4 categories: Objective, Methods, Results, and Conclusions. To evaluate their approach, they collected a corpus including 51,000 abstracts in which each sentence was explicitly assigned to one of the previous 4 categories by the authors. To describe each sentence, together with the surrounding ones, they used unigrams, bigrams and the relative location of the sentence inside the abstract.

5.2. *CHARACTERIZING SENTENCES*

We can characterize each sentence in a document by means of a wide range of semantic, syntactic, structural and positional features in order to evaluate how classification performance is affected by a varied, comprehensive set of information facets both generic and specific to scientific publications. The choice of our features was driven by the analysis of the most effective ones exploited by previous works. We exploited our feature-vector representation of sentences to compare the performance of four classification approaches: Naïve Bayes classifier, Support Vector Machine with linear kernel, Logistic Regression and Random Forest classifier. Note that the current library does not implement these methods and only relies on a Support Vector Machine algorithm, based on LIBSVM, for predicting the rhetorical category of a sentence. The nine groups of features we used to characterize the sentences of a paper are:

1. **Sentence length and position** (SENT_LP): the features of this group include shallow features describing the sentence and its position inside the document:
    a. *number of tokens of the sentence*: excluding stop words and tokens belonging to in-line citations;
    b. *length of the sentence*: as a nominal feature with values: Short (less than 15 tokens), Medium (from 15 to 30 tokens), Long (more than 30 tokens);
    c. *normalized position of the sentence inside the paper*: computed by dividing the paper in 10 folds of equal size. This feature represents the folder number from 1 to 10;

   d. *section number of the sentence*: the number of section in which the sentence occurs by assigning 1 to the abstract sentences and progressive integer numbers starting from 2 to sentences that belong to subsequent sections. We generated two features by considering only top level sections and by counting also nested sections;

   e. *normalized position of the sentence inside the section*: computed by dividing the section in 5 folds of equal size, it represents the folder number from 1 to 5. We generated two features respectively by considering the position in the top-level section and in the most nested section where the sentence occurs.

2. **Part-Of-Speech** (POS): these features are computed by relying on the POS tags of the tokens of each sentence:

   a. *top-level POSs*: we consider the percentage of tokens of the sentence belonging to each top-level POS category among the 15 top category of the Penn Treebank POS tagset[32]. The POS category of a POS tag is represented by its first character;

   b. *POS of the first verb*: one value among VBD (past tense), VBG (gerund or present participle), VBN (past participle), VBP (non-3rd person singular presen), VBZ (3rd person singular present) or NONE (no verbs in the sentence);

   c. *number of verbs by POS*: computed for each verb POS category considering the same set of categories of the previous feature;

   d. *number of comparative and superlative adjectives*: number of adjectives tagged with POS equal to JJR (comparative) and JJS (superlative).

3. **Dependency relations** (DEP_REL):

   a. *maximum depth of the dependency tree*;

   b. *number of edges of the dependency tree*;

   c. *number of edges of the dependency tree by edge type*: by considering the Penn Treebank Syntactic Dependencies tagset[28], we computed the percentage of arcs of the dependency tree belonging to each one of the 36 types of syntactic dependencies;

   d. *dependency relations unigrams*: for each ark of the dependency relation tree of a sentence we created a dependency relation unigram that is a token with the following structure: DEPRELtype_SOURCElemma_TARGETlemma. For instance the sentence 'We buy apples' would generate two unigrams: SBJ_we_buy, OBJ_apple_buy. We defined up to 400 boolean features per class to describe the presence in a sentence of the 400 dependency relations unigrams that are the most discriminative with respect to our classification task. We ignored dependency relations unigrams with frequency lower than 4. We set each feature equal to 1 if the related dependency relations unigram occurs in the sentence.

4. **Root verb** (R_VERB): we characterize the root verb of each sentence by specifying:

   a. *Form*: a nominal feature to indicate if the root verb is Active, Passive or there is no root verb in the sentence (NoVerb);

[32] https://catalog.ldc.upenn.edu/docs/LDC95T7/cl93.html

b. *Modal*: a nominal feature to indicate if the root verb is Modal or not (NoModal). This feature is equal to NoVerb if there is no root verb in the sentence;

c. *Tense*: a nominal feature that identifies if the tense of the root verb is Present, Future or Past. This feature is equal to NoVerb if there is no root verb in the sentence.

5. **Sentence similarity** (SENT_SIM): this group includes four features that are respectively equal to the tf*idf similarity of each sentence with respect to:

   a. *the title of the paper*;

   b. *the title of the top-level section* in which the sentence occurs;

   c. *the previous sentence*, if any in the same section;

   d. *the following sentence*, if any in the same section.

6. **Citations** (CITS): this group of features is useful to characterize the sentences in which citations occur and includes:

   a. *Number of citation spans*: number of groups of consecutive citations that occur in the text of the sentence. We compute three features to point out the number of citation spans in the sentence under analysis as well as in the previous and in the following sentence, if any in the same section;

   b. *Number of citation markers*: number of papers cited in the text of the sentence. We compute three features to point out the number of papers cited in the sentence under analysis as well as in the previous and in the following sentence, if any in the same section;

   c. *Number of syntactic citation spans*: number of groups of one or more consecutive citations that have a syntactic role in the sentence where they occur (representing for instance the subject of the sentence);

   d. *Position of the first citation*: this is a nominal feature with the following values: Beginning (first citation occurs in the first 20% of the sentence length), Middle (first citation occurs after the 20% and before the 80% of sentence length), End (first citation occurs in after the 80% of the sentence length), NoCit (if there are no citations in the sentence).

7. **Cue-based expressions** (CUE_EXP): this group of features is useful to detect the presence in the sentence of cue phrases or specific expressions:

   a. *Position of the first first-person pronoun, first third-person pronoun and first determiner*: this nominal features have one of the following values: Beginning (first 20% of the sentence length), Middle (after the 20% and before the 80% of sentence length), End (after the 80% of the sentence length), None (if there are no occurrences in the sentence).

   b. *Presence of contrary expressions*, from the list of 44 contrary expressions taken from [47];

   c. *Presence of speculations cues*:, from the list of 25 speculation cues identified by [46];

   d. *Presence of subjectivity cues*, considering the list of 8,222 (both strong and weak) subjectivity cues identified by [45];

   e. *Presence of negations*, considering the list of 32 negation expressions identified by [44] (Appendix D.4);

f. *Presence of verbs of a specific action lexicon verb group*, considering the 18 Action Lexicon verb groups identified by [44] (Appendix D.3). For each verb group we generated two boolean features that identify the presence in the sentence of one or more verbs of that group respectively in their negated and non-negated form.

8. **Section type** (SECT_TYPE): this is the only single-feature group. By means of a set of heuristics applied to the title of the top-level section in which the sentence to characterize occurs, we manage to associate to about 60% of the sentence of the DRI Corpus one section type in the following set: Abstract, Intro, Implementation, Background, Model, Description, Experiment, Result, Evaluation, Method, Algorithm, Discussion, Conclusions, FutureWork, Acknowledgements. To the sentences for which we were not able to determine a section type, we set this nominal feature equal to NoSectionType.
9. **N-grams and skip-n-grams** (N_GRAM_SK): this feature group includes the two following feature sets:
    a. *Unigrams, bigrams, trigrams*: by considering up to 200 most discriminative unigrams, bigrams and trigrams per class, this set of features identifies the presence of each one of these elements inside a sentence. We ignored unigrams, bigrams and trigrams with frequency lower than 4. To determine unigrams, bigrams and trigrams we used the lemmatized, lowercased tokens of sentences;
    b. *Skip1grams, skip2grams, skip3grams*: by considering up to 200 most discriminative skip1grams, skip2grams and skip3grams per class, this set of features identifies the presence of each one of these elements inside a sentence. We ignored skip1grams, skip2grams and skip3grams with frequency lower than 4. To determine skip1grams, skip2grams and skip3grams we used the lemmatized, lowercased tokens of sentences.

All the previous features were computed by relying on the results of the syntactic and semantic analysis performed by processing each paper thanks to the DRI library as described in Section 3. In Table 3, considering the sentences of the DRI Corpus, we show how many features are present for each feature group.

Features group	*Number of features*
SENT_LP	*7*
POS	*24*
DEP_REL	*1,694**
R_VERB	*3*
SENT_SIM	*4*
CITS	*7*
CUE_EXP	*72*
SECT_TYPE	*1*

N_GRAM_SK	*4,348**
Total features:	*6,160*

TABLE 3 - NUMBER OF FEATURES BY FEATURE GROUP, CONSIDERING THE SET OF RHETORICALLY ANNOTATED SENTENCES OF THE DRI CORPUS. FEATURE OCCURRENCE COUNTS MARKED BY * POINT OUT THE FEATURE GROUPS WITH A NUMBER OF FEATURES THAT DEPENDS ON THE TEXTUAL CONTENT OF THE CORPUS

5.3. *EVALUATION AND DISCUSSION*

We evaluated the performance of four classification approaches with respect to the identification of the rhetorical category of the sentences of the DRI Corpus (see Section 4.1). In our experiment we considered only the 5 top-level rhetorical categories manually associated to each sentence, leaving for future research the investigation of approaches aware of the 3 sub-categories.

Rhetorical category	Naïve Bayes	SVM	Logistic Regression	Random Forest
Background	0.620	0.680	0.714	0.613
Outcome	0.474	0.572	0.591	0.130
Challenge	0.308	0.431	0.355	0.009
Approach	0.649	0.808	0.831	0.761
Future Work	0.381	0.635	0.563	0.085
Unspecified	0.561	0.711	0.696	0.587
Weighted avg.	**0.591**	**0.721**	**0.737**	**0.581**

TABLE 4 - EVALUATION OF SENTENCE-BASED RHETORICAL CATEGORY CLASSIFIERS. F-1 SCORE RESULTING FROM THE 10-FOLD CROSS VALIDATION AGAINST THE GOLD STANDARD ANNOTATIONS OF THE DRI CORPUS

Table 4 shows the result of a 10-fold cross validation of the four classification methodologies we investigated, in terms of F-1 score. The Logistic Regression classifier obtains the best performances (F-1 equal to 0.74), immediately followed by the Support Vector Machine with linear kernel. We can notice that the DRI Corpus includes more than three times more sentences classified as Challenge (449) with respect to the ones classified as Future Work (136). Nevertheless, all the classifiers manage to identify Future Work sentences better than Challenge ones, with improvements of F-1 score of about 0.2 points if we consider the Logistic Regression or the Support Vector Machine. This fact can be explained by considering that when the future venues of research are presented in a paper, the authors usually rely on a set of linguistic traits and expressions that can be characterized in a more precise way than the case in which they present Challenges. For instance, when describing future works usually the future tense is used together with specific set of expressions like 'in the future', 'our future plans', etc. From Table 3 we can also notice that, because of the high number of attributes of the training instances (6,160) the Random Forest classifier is the approach with less discriminative power since it almost doesn't manage to correctly classify any element of the two classes with a lower number of training instances: Outcome and Future Work.

We investigated the contribution of each one of the nine groups of features described in Section 5.3 with respect to the adoption of a specific classifier. To this purpose, we evaluated each classifier by relying only on each specific feature group, as shown in Table 5. From Table 5, we can notice that three classification approaches over four obtain the best performance when trained on the *n-grams and skip-n-grams* (N_GRAM_SK) feature group. The Random Forest classifier instead gets its highest F-1 score when trained only on the *sentence length and position* (SENT_LP) features that is a small group of 7 relevant features (with respect to the N_GRAM_SK feature group that includes 4,348 features). This fact can be related to the ability of Random Forest to obtain good performance even with a small number of significant features. *Root verb* (R_VERB), including three nominal features describing the main verb of the sentence and *sentence similarity* (SENT_SIM) are the less effective feature groups when considered singularly (Table 5).

Features group considered	Naïve Bayes	SVM	Logistic Regression	Random Forest
SENT_LP	0.524	0.476	0.492	0.653
POS	0.435	0.386	0.479	0.497
DEP_REL	0.278	0.615	0.617	0.504
R_VERB	0.365	0.365	0.365	0.365
SENT_SIM	0.360	0.365	0.367	0.408
CITS	0.457	0.440	0.449	0.447
CUE_EXP	0.509	0.485	0.521	0.504
SECT_TYPE	0.529	0.529	0.529	0.528
N_GRAM_SK	0.586	0.621	0.636	0.551
All features	**0.591**	**0.721**	**0.737**	**0.581**

TABLE 5 - SINGLE FEATURE GROUP CONTRIBUTION TO CLASSIFICATION: F-1 SCORE OF EACH CLASSIFIER COMPUTED BY RELYING ON A 10-FOLD CROSS VALIDATION WHEN WE CONSIDER SEPARATELY EACH GROUP OF FEATURES. THE LAST LINE SHOWS THE F-1 SCORE OBTAINED BY CONSIDERING ALL FEATURES

Considering the most relevant sentence features in terms of information gain (Table 6) we can notice that the most represented features groups are *sentence length and position* (SENT_LP) and the *citations* (CITS). Both knowing where a sentence is positioned inside a paper as or inside the section where it occurs and considering the number and position of citations of sentences provide relevant information to determine its rhetorical category. Also the presence of pronoun inside the sentence and in particular of first-person ones constitute relevant classification features.

Features group	Feature name
SENT_LP	section number considering top-level headers
SENT_LP	normalized position of the sentence
SECT_TYPE	type of section
SENT_LP	section number considering nested headers
CUE_EXP	position of the first first-person pronoun
CITS	number of citation markers
CITS	position of first citation
CITS	number of citation spans
N_GRAM_SK	unigram we
POS	percentage of pronoun tokens

TABLE 6 - TOP-10 FEATURES IN TERMS OF INFORMATION GAIN.

6. *CITATION PURPOSE CLASSIFICATION*

The possibility to automatically characterize the semantics of citations would open a wide range of new possibilities with respect to the definition of new finer-grained metrics to evaluate research as well as the investigation of new patterns to navigate and search for scientific information. In this Section we present our experiments to evaluate the performance of different citation purpose classification approaches.

6.1. *RELATED WORK*

The automated identification of distinct semantic traits of citations has been explored by proposing several approaches. Most of these approaches model the context of each citation in order to determine the semantic category that better describes the same citation. Before the identification of a specific semantic category to characterize a citation, several studies investigated approaches to determine the context of a citation including all the textual excerpts that usually surround the citation and are useful to understand the reason why an article is cited. In particular, [19] found that Conditional Random Fields sequence labeller outperforms Support Vector Machine classifiers in identifying the sentences that belong to the context of a citation. [71] experimented with Markov Random Fields to automatically identify the context of citations. Besides the citation context, similarly to what happens with automated discursive annotation methodologies, the different approaches to citation characterization can be mainly distinguished with respect to the training corpus they consider, the feature set used to represent each citation and the algorithms exploited to determine semantic traits of the same citation.

[40] exploited a K-nearest neighbour classifier to associate a specific semantic facet to citations. They manually annotated 2,829 citations of a corpus of 116 articles by a schema that includes 4 top facets (weakness, positive, contrast, neutral) and 12 sub-facets. They represented each occurrence of a citation by means of a considerable number of features

including list of cue phrases, verb tense and modality, location of the citation and if the one considered is an auto-citation.

[72] used a corpus including 8,736 citations from 310 research papers of the ACL Anthology. Each citation was tagged manually as positive, negative or objective. They evaluated how a Support Vector Machine manages to determine the polarity of citations described by syntactic and semantic features including subjectivity cues and the occurrences of words for a science lexicon.

[73] annotated manually a corpus of 1,768 citations extracted from papers published in the ACL Anthology in 2007 and 2008. Each citation was assigned a category among: Background, Fundamental idea, Technical basis and Comparison. To describe each citation, they proposed features concerning their location and density in the sentence where they occur as well as in the surrounding ones. They evaluated the following classification approaches: BayesNet, Naïve Bayes, Support Vector Machine, J48 and RandomForest.

[74] performed its experiments with a corpus of 2,008 citations extracted from ACL Anthology papers from the year 2004. They manually annotated each citation by relying on the schema defined by [75]. To classify citations they reused features from [40], [73] and [72], extended with other ones spotting the presence of linguistic information like comparative or superlative adjectives or personal pronouns. They exploited a Maximum-Entropy classifier.

[76] annotated 91 biomedical articles from the Open Access subset of PubMed with 6,355 citations by a schema made of three top-classes: positive, neutral and negative in turn specialized by more specific sub-classes. They evaluated a Maximum-Entropy model by relying on features describing the lexical structure, the vocabulary and the placement of the citation inside the paper.

[19] exploited an annotations schema including 6 purpose classes and 4 polarity classes. The purpose and polarity of 3,500 citations extracted from papers of the ACL Anthology were manually annotated and three classifiers were compared: Support Vector Machine, Naïve Bayes and Logistic Regression.

[77] exploited a three-classes citation classification schema. Their approach to classify citation is particularly interesting since they exploited, besides lexical and linguistic textual features, also a set of features derived from the citation network each citation is part of, including the out-degree centrality of citing paper and the citation in-degree and out-degree centrality for 1st author of cited paper. They observed that textual features are more relevant in citation classification that network ones. Anyway, when it is impossible to access the citation context, the availability of citation network information could be useful to characterize citation.

6.2. *APPROACH*

In our experiments we treated citation purpose classification as a sentence classification tasks. We represented each sentence containing citations by means of the same set of features that we used to evaluate discursive sentence classification, described in detail in section 5. In our experiments we assume that the citation context is limited to the sentence in which the citation occurs. To know the number of features of each feature group, it is possible to refer to the data of Table 3, with the following two exceptions: the DEP_REL

features that are 940 and the N_GRAM_SK features are 2,352, since we are considering a token vocabulary smaller that in the case of rhetorical sentence classification (not all sentences of a paper include one or more in-line citations). Moreover, in our citation purpose classification experiments we considered an additional group of features besides the nine groups identified to support rhetorical sentence classification: the *Rhetorical Gold Standard category* (RHET_GS) of the citation sentence. Thanks to the peculiar structure of the DRI Corpus, including three layers of manual annotations over the same set of 40 papers, we can investigate how the annotations provided in one layer (rhetorical categories of sentences) help in the classification of other aspects annotated in different layers (citation purpose of sentences). Indeed, by considering the RHET_GS group of features, we can determine if and to what extent knowing the rhetorical category of a citing sentence helps in identifying its citation purpose. The RHET_GS group includes 6 features, one for each of the 5 top-level rhetorical categories plus the feature Unspecified that identifies sentences for which the annotator couldn't determine the rhetorical category (acknowledgements, meta-discourse, etc.). The value of each feature is equal to the percentage of annotators (over three) that classified that sentence of the corpus with the related rhetorical category.

The total number of features that characterize each citation sentence is 3,410. By exploiting this sentence representation, we evaluated four classification algorithms with respect to their ability to identify the citation purpose of the 2,356 citation sentences of the DRI Corpus.

6.3. *EVALUATION AND DISCUSSION*

In this section we discuss the performance of four classification approaches with respect to the identification of the purpose of the citation sentences of the DRI. We considered only the top-level citation purpose associated to each citation sentence (see Table 7).

Table 7 shows the performance of the four classifiers analyzed in identifying automatically the citation purpose of citation sentences by relying on the whole set of features to characterize each sentence; each classifier has been evaluated by means of a 1-fold-cross-validation over the Gold Standard annotations of the Dr. Inventor Multi-layered Corpus. We can notice that the Logistic Regression classifier obtains the best performance, with an F-1 score equal to 0.45. A baseline majority-class classifiers obtains an F-1 score of 0.247 by assigning to all the citation sentences the *Neutral* class that is the class with more training instances, 983 *Neutral* citing sentences over a total of 2,356 (see Table 2). The value of the best F-1 score (0.45 by relying on the Logistic Regression classifier) underlines the difficulty of identifying a set of linguistic and semantic features that enables a precise characterization of the purpose of citations.

With respect to each citation purpose, the performance of all classifiers is not always proportional to the number of training examples (i.e. annotated citation sentences). *Substantiation* is by far the citation purpose most difficult to identify with only 62 example citation sentences: the Random Forest classifier doesn't manage to classify correctly any *Substantiation* citation sentence. The *Substantiation* purpose should spot citations in which the cited paper and the citing paper support each other; this situation is infrequent and often difficult to identify for annotators. From Table 2 we can notice that the *Comparison* and *Basis* citation sentences have approximately the same number of instances in the DRI Corpus (211 and 201 respectively). On the contrary, the performance of all the classifiers is way better for the class *Comparison* than for the class *Basis*. This

could be motivated by the presence of linguistic traits that manages to characterize better the situation in which a citation is used to compare the own work to the cited one, rather than when a citation is used to spot the cited paper as one of the bases of the own work.

Citation purpose	Naïve Bayes	SVM	Logistic Regression	Random Forest
CRITICISM	0.497	0.435	0.442	0.342
USE	0.380	0.265	0.313	0.157
SUBSTANTIATION	0.074	0.042	0.088	0.000
COMPARISON	0.348	0.288	0.330	0.107
NEUTRAL	0.511	0.529	0.617	0.565
BASIS	0.200	0.130	0.096	0.000
Weighted avg.	0.438	0.403	0.450	0.352

TABLE 7 - EVALUATION OF SENTENCE-BASED CITATION PURPOSE CLASSIFIERS. F-1 SCORE RESULTING FROM THE 10-FOLD CROSS VALIDATION AGAINST THE GOLD STANDARD ANNOTATIONS OF THE DR. INVENTOR MULTI-LAYERED CORPUS.

In Table 8 we analyze the relevance of each feature group by measuring the performance of each classifier when we consider only the features of a specific group. From Table 8 we can notice that by relying only on the 6 features of the RHET_GS group, both the Random Forest and the Naïve Bayes classifiers, robustly dealing with small groups of features, manage to have good F-1 score. In particular, the Random Forest reaches its all-features F-1 score by considering only RHET_GS features. As a consequence we can state that the knowledge of the rhetorical category of a sentence contributes to identify its citation purpose.

Features group considered	Naïve Bayes	SVM	Logistic Regression	Random Forest
SENT_LP	0.302	0.246	0.258	0.294
POS	0.376	0.278	0.364	0.306
DEP_REL	0.345	0.370	0.383	0.325
R_VERB	0.251	0.246	0.246	0.249
SENT_SIM	0.256	0.246	0.245	0.230
CITS	0.347	0.346	0.347	0.339
CUE_EXP	0.434	0.400	0.410	0.323
SECT_TYPE	0.277	0.277	0.268	0.280
N_GRAM_SK	0.401	0.366	0.427	0.340

RHET_GS	0.387	0.316	0.332	0.357
All features	**0.438**	**0.403**	**0.450**	**0.352**

TABLE 8 - SINGLE FEATURE GROUP CONTRIBUTION TO CITATION PURPOSE CLASSIFICATION: F-1 SCORE OF EACH CLASSIFIER COMPUTED BY RELYING ON A 10-FOLD CROSS VALIDATION WHEN WE CONSIDER SEPARATELY EACH GROUP OF FEATURES.

In Table 9 we show a list of the top-10 features with respect to their information gain. By inspecting these features we can notice that pronouns play a relevant role with respect to the identification of the purpose of a citation. Half of the features shown in Table 9 are related to pronouns and four of them to first-person pronouns: the position and the presence of first-person pronouns and the presence of the pronouns 'we' and 'our'. The relevance of first-person pronouns could be related to the fact that in citation sentences with *Comparison* and *Use* purposes the authors of a paper explicitly mention their work and the approach they followed in order to compare to other studies or to mention other research outcomes they used. From Table 9 it is evident that the rhetorical category of a citation sentence is useful to determine its citation purpose. Indeed, the second and the third most relevant features with respect to information gain are the percentage of annotators (over three) that respectively tagged the citation sentence with the rhetorical category Background or Approach. As expected, among the top-10 features by information gain, there are two that characterize the citations that are present in the citation sentence: the position of the first citation in the sentence (Beginning, Middle, End) and the number of papers cited in the sentence.

Features group	Feature name
CUE_EXP	position of the first first-person pronoun
CUE_EXP	presence of a first-person pronoun
RHET_GS	percentage of annotators that tagged the sentence with the Background rhetorical category
RHET_GS	percentage of annotators that tagged the sentence with the Approach rhetorical category
POS	percentage of pronoun tokens
N_GRAM_SK	unigram our
CITS	position of first citation
SECT_TYPE	type of section
N_GRAM_SK	unigram we
CITS	number of citation markers

TABLE 9 - TOP-10 FEATURES IN TERMS OF INFORMATION GAIN. THE RHET_GS FEATURES ARE DERIVED FROM HUMAN ANNOTATIONS OF THE DRI CORPUS.

7. *CAUSALITY RELATION EXTRACTION*

We performed an explorative evaluation of the quality of the causal relations identified by the Causality spotter module. We randomly chose 10 Computer Graphics papers from the article presented at ACM SIGGRAPH Conferences between 2002 and 2015. The Causality spotter module was able to identify 157 causal relations inside the textual content of these papers.

We manually analyzed these causal relations to verify their correctness. We noted that the 74.5% of the causal relations extracted were **correct** (117 causal relations over 157). 29 causal relations over 157 were **partially correct**, thus having one among the cause and the effect wrongly marked. In 11 cases over 157 causal relations (7%), the causal relations identified were **incorrect** since the Causality spotter rules identified a causal relation in sentences where no causal relations were present. As a consequence, the **precision** of the Causality spotter module evaluated over this collection of 10 papers is 0.75 if we consider as valid ones all the causal relations for which both the cause and the effect have been correctly identified (strict precision). If we consider as valid causal relations the cases in which at least one entity among the cause and the effect has been correctly identified, the precision raises up to 0.93 (lenient precision). Since we did not manually annotate all the causal relations occurring in the 10 SIGGRAPH papers, we can't provide the recall of the Causality spotter over this collection of articles.

8. THE FRAMEWORK IN PRACTICE

After the detailed functional description of the scientific text analysis modules that are integrated in the Dr. Inventor Text Mining library provided in Section 2, in this Section we describe practical use-cases of the java library (DRI library) in order to extract structural, linguistic and semantic information from scientific publications. First of all we explain how to integrate the DRI library in an existing java project. Then we provide practical examples of how to use the library to carry out a varied set of analysis of scientific publications. All the information provided in this Section refers to the latest version of the DRI library at time of writing (version c.1.0.3). To access to the latest version of the DRI library, together with the related code examples and javadoc, the interested reader can access the Dr. Inventor Text Mining library web site: http://taln.upf.edu/pages/dri.upf/index.htm

8.1. IMPORTING THE LIBRARY

The Dr. Inventor Text Mining library is implemented as a self-contained Java library (referred to as DRI library) that can be easily integrated in any existing java project. Users can choose among two approaches to import the library:

- **Maven Dependency**: if Maven is used as the dependency manager, it is possible to import the DRI library by simply performing the following two modifications to the POM of the root Maven project the user is working with:
    - Addition of the following Maven repository:

```
<repositories>
        <repository>
        <id>backingdata-repo</id>
        <name>Backingdata repository</name>
<url>http://backingdata.org/dri/library/mavenRepo/</url>
        </repository>
</repositories>
```

- Addition of the DRI library dependency (in which the version 2.0 of the library is referred):

```
<dependency>
        <groupId>edu.upf.taln.dri</groupId>
        <artifactId>lib</artifactId>
        <version>c.1.0.3</version>
</dependency>
```

- **JAR Download**: it is possible to download the JAR file of the DRI library together with the all the dependent JAR files. To request more information on how to get the JAR version of the DRI library, please visit the web site: http://taln.upf.edu/pages/dri.upf/index.htm

Besides importing the JARs, in order to process a scientific publication by means of the DRI library it is needed to:

- download locally the ***DRI library configuration file***, accessible at the "/path/to/DRIconfig.properties". This file includes a set of name/value pairs that are useful to tune and customize the configuration parameters of the DRI library;
- download locally the ***DRI resources directory***, accessible at the DRI_RESOURCE_DIR_LOCAL_PATH. This directory groups the set of language resources, gazetteers and language models that the DRI library loads and exploits to process scientific articles. To request more information on how to get the *DRI resources directory* associated to a specific version of the DRI library, please visit the web site: http://taln.upf.edu/pages/dri.upf/index.htm (

Once the content of the *DRI resources directory* have been downloaded, we need to modify the *DRI library configuration file* as follows:

- set the value of the property **resourceFolder.fullPath** equal to the DRI_RESOURCE_DIR_LOCAL_PATH value (without a trailing file separator symbol / slash)

In order to perform all the analysis over the content of scientific publications, **the DRI library needs between 4 and 5 Gb of memory**. This requirement has to be taken into consideration by any Java program that imports the DRI library, since in most of the cases, for the DRI library to work, the heap size options of the Java Virtual Machine have to be properly tuned. In particular, it is necessary to set the maximum Java heap space equal or greater than 4,5 Gb (JVM option: -Xmx4500m).

Once the *DRI library configuration file* has been properly modified and enough Java heap space to process scientific publications by means of the DRI library has been provided, before starting to actually use the library, we need to initialize it. The initialization of the library is straightforward and consists of two steps:

- **specify the local path the *DRI library configuration file***, "/path/to/DRIconfig.properties". There are two approaches to specify this path:
  1. as a Java Virtual Machine argument named DRIpropertyFile:

     -DDRIpropertyFile="/path/to/DRIconfig.properties"

2. by passing the whole local path as the string argument of the following method:

   ```
   Factory.setDRIPropertyFilePath("/path/to/DRIconfig.properties");
   ```

- **invoke the following initialization method of the DRI library** that checks that all the settings are properly specified:

  ```
  Factory.initFramework();
  ```

  Such method call checks that the maximum Java heap allocated to the program is enough for the library to execute and that it is possible to access to the resources contained in the *DRI resources directory*. In case of any error, proper warnings are printed on the standard output and if fatal errors occur an exception is raised.

8.2. *EXPLOITING THE LIBRARY*

In this section we programmatically illustrate some common use-case scenario of use of the Dr. Inventor library. After describing the data model of scientific publication adopted by the DRI library, we describe how to use it to carry out the core scientific text mining tasks supported.

8.3. *THE SCIENTIFIC DOCUMENT DATA MODEL*

The DRI library defines a fully-fledged data model (set of classes) useful to represent the result of the varied set of text analyses that can be performed over a scientific publication. In particular, the information extracted thanks to the different modules of the DRI library can be accessed programmatically by using the methods exposed by the edu.upf.taln.dri.lib.model.Document interface. These methods support the retrieval of:

- the title and some information mined from the header of an article;
- the structure of sections, the list of sentences included in each section;
- the bibliographic entries eventually enriched with metadata from external web services;
- the SVO graph of a textual excerpt;
- the set of sentences selected by different extractive summarization approaches;
- the raw text of the article.

Each method of the edu.upf.taln.dri.lib.model.Document interface returns instances of objects of the Scientific Document data model useful to represent the data required. These data model object are useful to represent Sentences, Citations, Sections, etc.

By accessing the Javadoc on-line it is possible to obtain a detailed description of both the whole set of methods exposed by the edu.upf.taln.dri.lib.model.Document interface as well as the data objects included in the Scientific Document data model.

The first time that a method of the edu.upf.taln.dri.lib.model.Document interface is called, the time required to get the results is usually greater with respect to subsequent call because the actual processing action are executed and the related results computed. As explained in section 3, this is due to the default data processing strategy of the DRI library is based on a

lazy-processing approach: a specific text analysis is performed only the first time the results that analysis are required by the user.

8.4. *PROCESSING A PAPER*

Most of the scientific publication processing actions that can be performed by the DRI library are triggered by means of static methods exposed by the edu.upf.taln.dri.lib.Factory class. In this section we show how it is possible to import a paper to process, both in PDF and JATS XML format.

When the DRI library is initialized it is possible to enable or disable the different scientific text mining modules that it integrates. To this purpose, the `ModuleConfig` object has to be instantiated (see code example below). This object contains different boolean flags useful to manage the single scientific text mining modules of the Dr. Inventor Framework. If the related boolean flag is set to true, the scientific text mining module under consideration is activated and exploited to parse the scientific articles that are processed by mens of the DRI library.

The following code shows how to enable / disable the different modules of the DRI library programmatically:

```
// Instantiate the ModuleConfig class - the constructor sets all modules enabled by default
ModuleConfig modConfigurationObj = new ModuleConfig();

// Enable the parsing of bibliographic entries by means of online services (Bibsonomy, CrossRef, FreeCite, etc.)
modConfigurationObj.setEnableBibEntryParsing(true);

// Enable the parsing of the information from the header of the paper by means of online services (Bibsonomy, CrossRef, FreeCite, etc.)
modConfigurationObj.setEnableHeaderParsing(true);

// Enable the dependency parsing of the sentences of a paper
modConfigurationObj.setEnableGraphParsing(true);

// Enable coreference resolution
modConfigurationObj.setEnableCoreferenceResolution(true);

// Enable the extraction of causal relations
modConfigurationObj.setEnableCausalityParsing(true);

// Enable the association of a rhetorical category to the sentences of the paper
modConfigurationObj.setEnableRhetoricalClassification(true);

// Improt the configuration parameters set in the ModuleConfig instance
Factory.setModuleConfig(modConfigurationObj);
```

The following code is useful to check (print on the standard output) which modules are currently enabled:

```
System.out.println("Modules' enable status: " + Factory.getModuleConfig().toString());
```

Let suppose that we have a PDF paper to process that is locally stored at the path: /my/file/path/PDF_file_name.pdf. In order to import the paper we need to execute the following line of code:

```
Document doc_PDFpaperFILE =
Factory.getPDFloader().parsePDF("/my/file/path/PDF_file_name.pdf
");
```

The textual content is extracted from the PDF article and an instance of scientific publication implementing the edu.upf.taln.dri.lib.model.Document interface is returned.

Similarly, we can also download and import a PDF from an URL:

```
Document doc_PDFpaperURL = Factory.getPDFloader().parsePDF(new
URL("http://www2007.org/workshops/paper_45.pdf"));
```

To import a paper available as a file in JATS XML format at the path: /my/file/path/JATS_XML_file_name.xml, it is possible to invoke the following method:

```
Document doc_JATSpaperFILE =
Factory.getJATSloader().parseJATS("/my/file/path/JATS_XML_file_n
ame.xml");
```

The DRI library gives users the possibility to process directly a textual excerpt by means of the method:

```
Document doc_PlainText =
Factory.getPlainTextLoader().parsePlainText(textExcerpt,
textName);
```

where the first argument (String, **`textExcerpt`**) is the text excerpt to parse, while the second argument (String, **`textName`**) is an optional text title.

Since each document loaded in the main memory needs a considerable amount of memory, in order to free the memory from all the data resulting from the analysis of the content of a scientific publication, it is needed to invoke the cleanUp() method implemented by the edu.upf.taln.dri.lib.model.Document interface:

```
doc_PDFpaperFILE.cleanUp();
```

It is suggested to process collections of documents / papers by loading each document / paper individually, executing the processing tasks and then call the cleanUp() method.

8.5. *GENERATING SVO GRAPHS*

The graph of a text excerpt / scientific paper is a directed graph where each node is a single or multi token expression (nominal, pronominal or verbal) occurring in the same text excerpt and each arc is a relation of one of the following three types:

- **SUBJECT**: from a subject node (usually nominal) to a verb node;
- **OBJECT**: from an object node (usually nominal) to a verb node;
- **CAUSE**: from a cause node to an effect node. Both cause and effect node can be of any type: nominal, pronominal or verbal.

As a consequence, if we consider the sentence:

The proposed method considers the relationships among rigid body parts and is more general since the equation can handle motions of close interactions with/without tangles.

the DRI library will generate the graph shown in Figure 14.

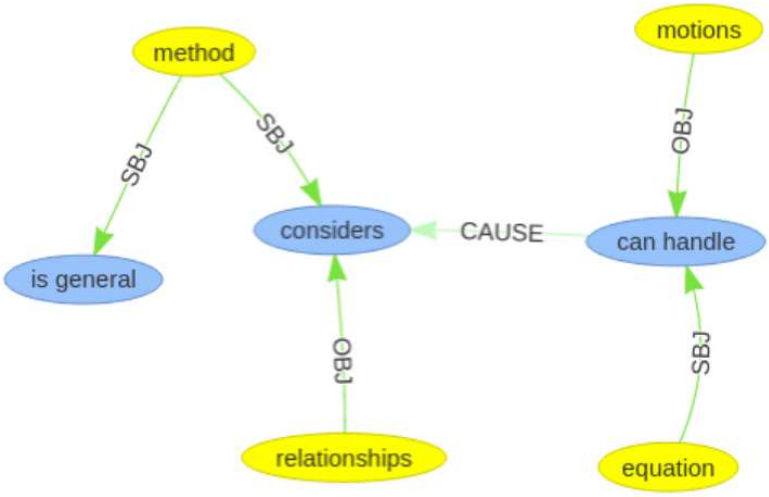

FIGURE 14 - GRAPH FOR A TEXT

In the graph shown in Figure 14, in yellow there are nominal nodes, while in blue verbal nodes. We can see that the verbal nodes 'is general' and 'can handle' are multi token expressions while all other nodes are composed of a single token. In this case the CAUSE relation is among two verbs since the fact that "the equation can handle motions" causes the following effect: "method considers relationships".

Each sentence generates parts of the graph of a whole textual excerpt or paper. Thanks to the coreference resolution process nominal and pronominal nodes of the graph that refer to the same entity are merged (both inside the same sentence and across different sentences). For instance, given the sentence:

Since kinematic constraints can usually be represented by single equations, they can be easily embedded into optimization problems for motion synthesis.

the related graph generated by the DRI library is shown in Figure 15.

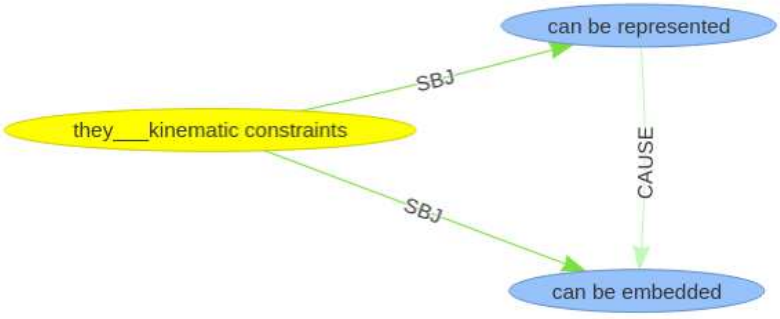

FIGURE 15 - SVO / ROS GRAPH EXAMPLE

In the graph shown in Figure 15, we can see that the nodes 'kinematic constraints' and 'they' have been merged by the coreference resolutor, since 'they' has been identified as a pronoun of 'kinematic constraints'. A single nominal node has is present in the ROS - the label of this nominal node is the concatenation of the labels of the merged nodes ('kinematic constraints' and 'they') separated by three '_' characters, thus resulting in 'they___kinematic constraints'.

As a consequence the nodes of a graph can:

- be derived from a single token or a multi token expression that occurs in a specific sentence ('can be embedded' node in the example above);
- be derived by merging one or more single token / multi token expressions (coreference nodes), each one coming from a specific sentence. This second kind of nodes is generated thanks to the coreference resolutor that suggests the aggregation of nodes from the same or different sentences, like the 'they___kinematic constraints' node in the example above.

Each multi token expression node is characterized by a head word that is the token of that expression that is most relevant to represent the ROS node. For instance, the following sentence:

Further, these methods cannot handle close interactions without any tangles.

generates the graph shown in Figure 16 that has one single token expression node ('methods') and two multi token expression nodes: 'close interactions' and 'Further can not handle'. The head word of the 'close interactions' node is 'interactions' while the head word of the 'Further can not handle' node is the model veb 'can'. Especially with nominal nodes, the head word is useful to point out the main noun of the node. In case a node is the result of the merging of two or more nodes thanks to coreference resolution, its head word will be a list of the head word of the merged nodes. For instance, in case of the node 'they___kinematic constraints' the head words will be both the word 'constraints' and the word 'they' (one for each merged node).

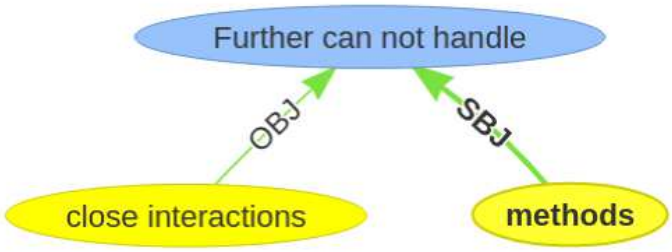

FIGURE 16 - GRAPH EXAMPLE

The DRI library supports the serialization of a graph of a scientific publication by means of the following two CSV tables:

- the **Document ROS CSV**, generated by invoking the static method:

edu.upf.taln.dri.lib.model.util.DocParse.**getDocumentROSasCSVstring**(Document doc, SentenceSelectorENUM sentenceSelector)

Each row of the The **Document CSV** describes an arc of the graph, by providing the name of the arc (edge), the id of the source and target nodes and other information describing the pair of nodes that compose the arc, including:

- the node names;
- the head word of the node (or list of head words separated by a comma in case of coreference nodes that are derived by merging all the coreferent nodes);
- the rhetorical class of the sentence the node belongs to (or comma-separated list of the rhetorical classes of all the coreferent nodes merged, when we consider a coreference node);
- the ID of the sentence that contains the node (empty if the node is a coreference node);
- the position of the token(s) of the node in the sentence with ID (comma-separated list of integer - first sentence token is at position 0 and so on - empty if the node is a coreference node).

- the **Sentence CSV**, generated by invoking the method:

edu.upf.taln.dri.lib.model.util.DocParse.**getSentencesCSVstring**(Document doc, SentenceSelectorENUM sentenceSelector)

Each row of the **Sentence CSV** describes a sentence of the article. For each sentence, the following information is specified:

- the sentenceID;
- the space-separated list of tokens of the sentence;
- the rhetorical class of the sentence identified by means of the Rhetorical annotator module;
- the name / title and the nesting level of the section of the document in which the sentence occurs;
- a Boolean flag to point out if the sentence includes or not an inline citation.

The **Sentence CSV** is useful to retrieve the text and metadata of the sentences of the processed document by using the sentence IDs that are present in each row of the **DocumenCSV**.

To get more information on both the **Document CSV** and the **Sentence CSV**, as well as to see practical examples of Document and Sentence CSV files generated from textual excerpts of a scientific publication, it is possible to browse the on-line Javadoc describing the class edu.upf.taln.dri.lib.model.util.DocParse[33].

8.6. *Persisting the processing results of a paper*

Once a paper is loaded in memory thanks to the DRI library, it is converted from its original format (PDF or JATS XML) to the Dr. Inventor document format and eventually processed. It is possible to persist the processing results of a paper so as to access them at a later time without the need to reload the original document (PDF or JATS XML) and reprocess it. Each processed paper can be serialized and stored as an XML file so as to be able to reload its content.

[33] http://backingdata.org/dri/library/c.1.0.3/doc/edu/upf/taln/dri/lib/model/util/DocParse.html

The code snippet below shows how to load a paper by converting a PDF file locally stored at the path: /my/file/path/PDF_file_name.pdf. Then the Dr. Inventor document is stored as the XML file named ***XML_paper_file.xml***.

```
Document doc_PDFpaperFILE =
Factory.getPDFloader().parsePDF("/my/file/path/PDF_file_name.pdf
");

Writer out = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream("XML_paper_file.xml"), "UTF-8"));

try {

      out.write(aString);

} finally {

      out.close();

}
```

Later, it is possible to instantiate the paper as a Document in the java code, from the XML file ***XML_paper_file.xml***, without converting the original PDF and without executing again the content analysis that were already performed at the moment of XML storage. The following code snippet shows how to load the XML file:

```
Document docLoadedFromSerializedXML =
Factory.createNewDocument("/my/file/path/XML_paper_file.xml");
```

9. *CONCLUSIONS*

In this document we presented the Dr. Inventor Text Mining library, a java library useful to analyze scientific texts. The library constitutes, to the best of our knowledge, one of the most rich and comprehensive text analysis libraries tailored to process scientific publications. We explained into details the features of the scientific text analysis modules integrated in the library by providing practical examples of use; we also evaluated the performance of the most relevant modules.

Besides the automated identification of the structural elements of scientific publications, the Dr. Inventor Text Mining library enables a varied set of fine-grained semantic analyses of the contents of a paper. These analyses include the characterization of the scientific discourse of publications by determining the rhetorical category of sentences, the identification of the purpose of citations, the generation of extractive summaries of papers and the representation of the content of a publication by means of SVO graphs. The evaluation of the performances of the semantic analyses presented in this manual demonstrates that, even if there is still room for improvement, the Dr. Inventor Text Mining library obtains competitive results. All the analysis of scientific publications described in this manual can be easily performed by anyone by importing the DRI java library that integrates in an coherent pipeline all the text mining tools presented.

BIBLIOGRAPHY

[1] Cunningham, Hamish, Valentin Tablan, Angus Roberts, and Kalina Bontcheva. "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics." PLoS Comput Biol 9, no. 2 (2013): e1002854.

[2] The Rise of Open Access. Science, Vol. 342 no. 6154 pp. 58-59 - https://www.sciencemag.org/content/342/6154/58.full (2013)

[3] Björk, Bo-Christer, Mikael Laakso, Patrik Welling, and Patrik Paetau. "Anatomy of green open access." Journal of the Association for Information Science and Technology 65, no. 2 (2014): 237-250.

[4] Solomon, David J., Mikael Laakso, and Bo-Christer Björk. "A longitudinal comparison of citation rates and growth among open access journals."Journal of informetrics 7, no. 3 (2013): 642-650.

[5] Laakso, Mikael, and Bo-Christer Björk. "Anatomy of open access publishing: a study of longitudinal development and internal structure." BMC medicine10, no. 1 (2012): 1.

[6] Lewis, David W. "The inevitability of open access." College & Research Libraries 73, no. 5 (2012): 493-506.

[7] Huh, Sun. "Coding practice of the Journal Article Tag Suite extensible markup language." Science Editing 1, no. 2 (2014): 105-112.

[8] Erkan, Günes, and Dragomir R. Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." *Journal of Artificial Intelligence Research* 22 (2004): 457-479.

[9] Tkaczyk, Dominika, Pawel Szostek, Piotr Jan Dendek, Mateusz Fedoryszak, and Lukasz Bolikowski. "CERMINE--Automatic Extraction of Metadata and References from Scientific Literature." In Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on, pp. 217-221. IEEE, 2014.

[10] Ramakrishnan, Cartic, Abhishek Patnia, Eduard Hovy, and Gully APC Burns. "Layout-aware text extraction from full-text PDF of scientific articles."Source code for biology and medicine 7, no. 1 (2012): 1.

[11] Clark, Christopher, and Santosh Divvala. "PDFFigures 2.0: Mining Figures from Research Papers." In Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, pp. 143-152. ACM, 2016.

[12] Peng, Fuchun, and Andrew McCallum. "Information extraction from research papers using conditional random fields." Information processing & management 42, no. 4 (2006): 963-979.

[13] Do, Huy Hoang Nhat, Muthu Kumar Chandrasekaran, Philip S. Cho, and Min Yen Kan. "Extracting and matching authors and affiliations in scholarly documents." In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries, pp. 219-228. ACM, 2013.

[14] Councill, Isaac G., C. Lee Giles, and Min-Yen Kan. "ParsCit: an Open-source CRF Reference String Parsing Package." In LREC, vol. 8, pp. 661-667. 2008.

[15] Luong, Minh-Thang, Thuy Dung Nguyen, and Min-Yen Kan. "Logical structure recovery in scholarly articles with rich document features." Multimedia Storage and Retrieval Innovations for Digital Library Systems 270 (2012).

[16] Liakata, Maria, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. "Automatic recognition of conceptualization zones in scientific articles and two life science applications." Bioinformatics28, no. 7 (2012): 991-1000.

[17] Teufel, Simone. "The Structure of Scientific Articles: Applications to Citation Indexing and Summarization (Center for the Study of Language and Information-Lecture Notes)." (2010).

[18] Nakov, Preslav I., Ariel S. Schwartz, and Marti Hearst. "Citances: Citation sentences for semantic analysis of bioscience text." In Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics, pp. 81-88. 2004.

[19] Abu-Jbara, Amjad, Jefferson Ezra, and Dragomir R. Radev. "Purpose and Polarity of Citation: Towards NLP-based Bibliometrics." In HLT-NAACL, pp. 596-606. 2013.

[20] Abu-Jbara, Amjad, and Dragomir Radev. "Coherent citation-based summarization of scientific papers." In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 500-509. Association for Computational Linguistics, 2011.

[21] Ronzano, Francesco, and Horacio Saggion. "An Empirical Assessment of Citation Information in Scientific Summarization." In International Conference on Applications of Natural Language to Information Systems, pp. 318-325. Springer International Publishing, 2016.

[22] Smit, Eefke, and Maurits Van Der Graaf. "Journal article mining: the scholarly publishers' perspective." Learned Publishing 25, no. 1 (2012): 35-46.

[23] Ciancarini, Paolo, Angelo Di Iorio, Andrea Giovanni Nuzzolese, Silvio Peroni, and Fabio Vitali. "Semantic annotation of scholarly documents and citations." In Congress of the Italian Association for Artificial Intelligence, pp. 336-347. Springer International Publishing, 2013.

[24] Sateli, Bahar, and René Witte. "What's in this paper?: Combining Rhetorical Entities with Linked Open Data for Semantic Literature Querying." InProceedings of the 24th International Conference on World Wide Web, pp. 1023-1028. ACM, 2015.

[25] Shotton, David. "Semantic publishing: the coming revolution in scientific journal publishing." Learned Publishing 22, no. 2 (2009): 85-94.

[26] Vahdati, Sahar, Dimou, Anastasia, Lange, Christoph and Di Iorio, Angelo "Semantic Publishing Challenge: Bootstrapping a Value Chain for Scientific Data" In Proceedings of the Semantics, Analytics, Visualisation: Enhancing Scholarly Data Workshop co-located with the 25th International World Wide Web Conference (2016)

[27] Thakker, Dhaval, Taha Osman, and Phil Lakin. "Gate jape grammar tutorial."Nottingham Trent University, UK, Phil Lakin, UK, Version 1 (2009).

[28] Abu-Jbara, Amjad, and Dragomir Radev. "Reference scope identification in citing sentences." In Proceedings of the 2012 Conference of the North American Chapter of the

Association for Computational Linguistics: Human Language Technologies, pp. 80-90. Association for Computational Linguistics, 2012.

[29] Nivre, J. (2003). An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*, Nancy, France, 23-25 April 2003, pp. 149-160..

[30] Fisas, Beatriz, Francesco Ronzano, and Horacio Saggion. "On the Discoursive Structure of Computer Graphics Research Papers." In The 9th Linguistic Annotation Workshop held in conjuncion with NAACL 2015, p. 42. 2015.

[31] Fisas, Beatriz, Francesco Ronzano, and Horacio Saggion. "A Multi-Layered Annotated Corpus of Scientific Papers. " In The Language Resource and Evaluation Conference (2016)

[32] Witten, Ian H., and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.

[33] Moro, Andrea, Francesco Cecconi, and Roberto Navigli. "Multilingual word sense disambiguation and entity linking for everybody." In Proceedings of the 2014 International Conference on Posters & Demonstrations Track-Volume 1272, pp. 25-28. CEUR-WS. org, 2014.

[34] Ronzano, Francesco, Beatriz Fisas, Gerard Casamayor del Bosque, and Horacio Saggion. "On the automated generation of scholarly publishing linked datasets: the case of CEUR-WS proceedings." In Semantic Web Evaluation Challenge, pp. 177-188. Springer International Publishing, 2015.

[35] Peroni, Silvio. "The Semantic Publishing and Referencing Ontologies." InSemantic Web Technologies and Legal Scholarly Publishing, pp. 121-193. Springer International Publishing, 2014.

[36] Ruiz-Iniesta, Almudena, and Oscar Corcho. "A review of ontologies for describing scholarly and scientific documents." In SePublica. 2014.

[37] Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task." In Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pp. 28-34. Association for Computational Linguistics, 2011.

[38] Saggion, Horacio. "A robust and adaptable summarization tool." Traitement Automatique des Langues 49, no. 2 (2008).

[39] Piwowar, Heather. "Altmetrics: Value all research products." Nature 493, no. 7431 (2013): 159-159.

[40] Teufel, Simone, Advaith Siddharthan, and Dan Tidhar. "Automatic classification of citation function." In Proceedings of the 2006 conference on empirical methods in natural language processing, pp. 103-110. Association for Computational Linguistics, 2006.

[41] Jaidka, Kokil, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. "Overview of the CL-SciSumm 2016 Shared Task." In Proc. of the Joint Workshop on Bibliometricenhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2016). 2016.

[42] Jaidka, Kokil, Muthu Kumar Chandrasekaran, Beatriz Fisas Elizalde, Rahul Jha, Christopher Jones, Min-Yen Kan, Ankur Khanna et al. "The computational linguistics summarization pilot task." Proceedings of TAC(2014).

[43] Teufel, Simone, and Marc Moens. "Summarizing scientific articles: experiments with relevance and rhetorical status." Computational linguistics28, no. 4 (2002): 409-445.

[44] Teufel, Simone. "Argumentative zoning: Information extraction from scientific text." PhD diss., University of Edinburgh, 2000.

[45] Riloff, Ellen, and Janyce Wiebe. "Learning extraction patterns for subjective expressions." In Proceedings of the 2003 conference on Empirical methods in natural language processing, pp. 105-112. Association for Computational Linguistics, 2003.

[46] Velldal, Erik, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. "Speculation and negation: Rules, rankers, and the role of syntax." Computational linguistics 38, no. 2 (2012): 369-410.

[47] Biber, Douglas. Variation across speech and writing. Cambridge University Press, 1991.

[48] Edmundson, Harold P. "New methods in automatic extracting." Journal of the ACM (JACM) 16, no. 2 (1969): 264-285.

[49] Kupiec, Julian, Jan Pedersen, and Francine Chen. "A trainable document summarizer." In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 68-73. ACM, 1995.

[50] Saggion, Horacio, and Thierry Poibeau. "Automatic text summarization: Past, present and future." In Multi-source, Multilingual Information Extraction and Summarization, pp. 3-21. Springer Berlin Heidelberg, 2013.

[51] Hovy, Eduard, and Chin-Yew Lin. "Automated text summarization and the SUMMARIST system." In Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998, pp. 197-214. Association for Computational Linguistics, 1998.

[52] Seki, Yohei. "Sentence Extraction by tf/idf and position weighting from Newspaper Articles." (2002).

[53] Halliday, Michael Alexander Kirkwood, and Ruqaiya Hasan. Cohesion in english. Routledge, 2014.

[54] Luhn, Hans Peter. "The automatic creation of literature abstracts." IBM Journal of research and development 2, no. 2 (1958): 159-165.

[55] Nenkova, Ani, and Lucy Vanderwende. "The impact of frequency on summarization." Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101 (2005).

[56] Saggion, Horacio, and Guy Lapalme. "Generating indicative-informative summaries with sumUM." Computational linguistics 28, no. 4 (2002): 497-526.

[57] Paice, Chris D., and Paul A. Jones. "The identification of important concepts in highly structured technical papers." In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 69-78. ACM, 1993.

[58] Abdalla, Rashid M., and Simone Teufel. "A bootstrapping approach to unsupervised detection of cue phrase variants." In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 921-928. Association for Computational Linguistics, 2006.

[59] Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing order into texts." Association for Computational Linguistics, 2004.

[60] Radev, Dragomir R., Hongyan Jing, and Malgorzata Budzikowska. "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies." In Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, pp. 21-30. Association for Computational Linguistics, 2000.

[61] Saggion, Horacio, and Robert Gaizauskas. "Multi-document summarization by cluster/profile relevance and redundancy removal." In Proceedings of the Document Understanding Conference, pp. 6-7. 2004.

[62] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out: Proceedings of the ACL-04 workshop, vol. 8. 2004.

[63] Saggion, Horacio. "Creating Summarization Systems with SUMMA." InLREC, pp. 4157-4163. 2014.

[64] Feltrim, Valéria D., Simone Teufel, Maria Graças V. das Nunes, and Sandra M. Aluísio. "Argumentative zoning applied to critiquing novices' scientific abstracts." In Computing Attitude and Affect in Text: Theory and Applications, pp. 233-246. Springer Netherlands, 2006.

[65] Guo, Yufan, Anna Korhonen, and Thierry Poibeau. "A weakly-supervised approach to argumentative zoning of scientific documents." In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 273-283. Association for Computational Linguistics, 2011.

[66] Guo, Yufan, Ilona Silins, Ulla Stenius, and Anna Korhonen. "Active learning-based information structure analysis of full scientific articles and two applications for biomedical literature review." Bioinformatics 29, no. 11 (2013): 1440-1447.

[67] Merity, Stephen, Tara Murphy, and James R. Curran. "Accurate argumentative zoning with maximum entropy models." In Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries, pp. 19-26. Association for Computational Linguistics, 2009.

[68] Mullen, Tony, Yoko Mizuta, and Nigel Collier. "A baseline feature set for learning rhetorical zones using full articles in the biomedical domain." ACM SIGKDD Explorations Newsletter 7, no. 1 (2005): 52-58.

[69] Mizuta, Yoko, and Nigel Collier. "An Annotation Scheme for a Rhetorical Analysis of Biology Articles." In LREC, pp. 1737-1740. 2004.

[70] Hirohata, Kenji, Naoaki Okazaki, Sophia Ananiadou, Mitsuru Ishizuka, and Manchester Interdisciplinary Biocentre. "Identifying Sections in Scientific Abstracts using Conditional Random Fields." In IJCNLP, pp. 381-388. 2008.

[71] Qazvinian, Vahed, and Dragomir R. Radev. "Identifying non-explicit citing sentences for citation-based summarization." In Proceedings of the 48th annual meeting of the

association for computational linguistics, pp. 555-564. Association for Computational Linguistics, 2010.

[72] Athar, Awais. "Sentiment analysis of citations using sentence structure-based features." In Proceedings of the ACL 2011 student session, pp. 81-87. Association for Computational Linguistics, 2011.

[73] Dong, Cailing, and Ulrich Schäfer. "Ensemble-style Self-training on Citation Classification." In IJCNLP, pp. 623-631. 2011.

[74] Jochim, Charles, and Hinrich Schütze. "Towards a generic and flexible citation classifier based on a faceted classification scheme." (2012).

[75] Moravcsik, Michael J., and Poovanalingam Murugesan. "Some results on the function and quality of citations." Social studies of science 5, no. 1 (1975): 86-92.

[76] Li, Xiang, Yifan He, Adam Meyers, and Ralph Grishman. "Towards Fine-grained Citation Function Classification." In RANLP, pp. 402-407. 2013.

[77] Xu, Han, Eric Martin, and Ashesh Mahidadia. "Using heterogeneous features for scientific citation classification." In Proceedings of the 13th conference of the Pacific Association for Computational Linguistics. 2013.