Universidade de Aveiro Departamento de
2014 Eletrónica, Telecomunicações e Informática

Jóni Amauri de Almeida Lourenço

Plataforma colaborativa de anotação de literatura biomédica

A web-based collaborative curation system for biomedical literature

Universidade de Aveiro Departamento de
2014 Eletrónica, Telecomunicações e Informática

Jóni Amauri de Almeida Lourenço

Plataforma colaborativa de anotação de literatura biomédica

A web-based collaborative curation system for biomedical literature

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor José Luís Oliveira, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Sérgio Aleixo Matos, Investigador Auxiliar do Instituto de Engenharia Eletrónica e Telemática de Aveiro.

Aos meus pais.

o júri / the jury

presidente / president	Prof. Doutor Augusto Silva Professor Auxiliar do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro
vogais / examiners committee	Prof. Doutor Sérgio Sobral Nunes Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto
	Prof. Doutor José Luís Oliveira Professor Associado do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro (orientador)

agradecimentos / acknowledgements

Gostava de agradecer, em primeiro lugar, aos meus orientadores, José Luís Oliveira e Sérgio Matos, principalmente pela oportunidade e orientação, mas também pelo apoio incondicional durante o mestrado. Os seus conselhos, ensinamentos e recomendações foram vitais para a execução deste trabalho. Agradeço também aos membros do Grupo de Bioinformática pela constante ajuda e conversas divertidas, que o tornaram num local acolhedor para estar e trabalhar.

Um agradecimento especial ao David Campos não só pela incansável ajuda e amizade mas também pelas longas discussões e ensinamentos que contribuíram para a realização deste trabalho. Foi um prazer poder trabalhar contigo.

Gostava também de agradecer aos meus amigos, Anton Zverev, Renato Pinho, Michael Domingues e Luís Ribeiro pelas excelentes conversas, momentos de convívio e paciência infinita e igualmente ao Hugo Araújo, Luís Lemos e Catarina Novo pelo apoio e amizade incondicional.

Aproveito ainda para agradecer a todos aqueles que ao longo destes anos contribuíram, direta ou indiretamente, para a realização desta tese, nomeadamente aos meus colegas do curso Mestrado Integrado em Engenharia de Computadores e Telemática e a todos aqueles com quem tive o privilégio de trabalhar na Associação Académica da Universidade de Aveiro. Com certeza que muitos foram os ensinamentos que me transmitiram que ajudaram na realização deste trabalho.

Agradeço também ao meu irmão, Hugo Lourenço, pelo constante apoio e amizade que tornaram possível a realização deste trabalho.

Por último, mas com igual importância, agradeço aos meus pais Sebastião Lourenço e Elisabete Lourenço, por me possibilitarem esta oportunidade, por todo o apoio, amizade, amor e educação que me deram ao longo destes anos. A eles dedico esta dissertação.

palavras-chave

Bioinformática, mineração de texto, mineração interactiva, anotação de documentos biomédicos, extração de informação, reconhecimento de conceitos.

resumo

Com o acréscimo da quantidade de literatura biomédica a ser produzida todos os dias, vários esforços têm sido feitos para tentar extrair e armazenar de forma estruturada os conceitos e as relações nela presentes. Por outro lado, uma vez que a extração manual de conceitos compreende uma tarefa extremamente exigente e exaustiva, algumas soluções de anotação automática foram surgindo. No entanto, mesmo os sistemas de anotação mais completos não têm sido muito bem recebidos no seio das equipas de investigação, em grande parte devido às falhas a nível de usabilidade e de interface *standards*. Para colmatar esta falha são necessárias ferramentas de anotação interativa, que tirem proveito de sistemas de anotação automática e de bases de dados já existentes, para ajudar os anotadores nas suas tarefas do dia-a-dia.

Nesta dissertação iremos apresentar uma plataforma de anotação de literatura biomédica orientada para a usabilidade e que suporta anotação manual e automática. No mesmo sentido, integramos no sistema várias bases de dados, no intuito de facilitar a normalização dos conceitos anotados. Por outro lado, os utilizadores podem também contar com funcionalidades colaborativas em toda a aplicação, estimulando assim a interação entre os anotadores e, desta forma, a produção de melhores resultados. O sistema apresenta ainda funcionalidades para importar e exportar ficheiros, gestão de projetos e diretivas de anotação.

Com esta plataforma, Egas, participámos na tarefa de anotação interativa do BioCreative IV (IAT), nomeadamente na identificação de interações proteína-proteína. Depois de avaliado por um conjunto de anotadores, o Egas obteve os melhores resultados entre os sistemas apresentados, relativamente à usabilidade, confiança e desempenho.

keywords

Bioinformatics, text mining, interactive mining, biomedical document curation, information extraction, concept recognition.

abstract

With the overwhelming amount of biomedical textual information being produced, several manual curation efforts have been set up to extract and store concepts and their relationships into structured resources. Since manual annotation is a very demanding and expensive task, computerized solutions were developed to perform such tasks automatically. Nevertheless, high-end information extraction techniques are still not widely used by biomedical research communities, mainly due to the lack of standards and limitations in usability. Interactive annotation tools intend to fill this gap, taking advantage of automatic techniques and existing knowledge bases to assist expert curators in their daily tasks.

This thesis presents Egas, a web-based platform for biomedical text mining and assisted curation with highly usable interfaces for manual and automatic inline annotation of concepts and relations. Furthermore, a comprehensive set of knowledge bases are integrated and indexed to provide straightforward concept normalization features. Additionally, curators can also rely on real-time collaboration and conversation functionalities allowing discussing details of the annotation task as well as providing instant feedback of curators interactions. Egas also provides interfaces for on-demand management of the annotation task settings and guidelines, and supports standard formats and literature services to import and export documents. By taking advantage of Egas, we participated in the BioCreative IV interactive annotation task, targeting the assisted identification of protein-protein interactions described in PubMed abstracts related to neuropathological disorders. Thereby, when evaluated by expert curators, Egas obtained very positive scores in terms of usability, reliability and performance. These results, together with the provided innovative features, place Egas as a state-of-the-art solution for fast and accurate curation of information, facilitating the task of creating and updating knowledge bases in a more consistent way.

Contents

Contents i

List of Figures iii

List of Tables vii

Introduction 1

1.1. Motivation 4
1.2. Results 5
1.3. Thesis outline 5

Biomedical information extraction 7

2.1. Resources 10
2.2. Pre-processing 11
2.3. Concept recognition 14
2.3.1. Resources 15
2.3.2. Named entity recognition 16
2.3.3. Normalization and disambiguation 18
2.4. Relation Mining 19
2.4.1. Resources 22
2.4.2. Document classification 22
2.4.3. Trigger recognition 23
2.4.4. Relation extraction 23
2.5. Interactive Mining 24
2.6. Evaluation 29
2.7. Summary 30

Requirements 31

3.1. User requirements 31
3.1.1. Mission 32
3.2. Functional requirements 32
3.3. Non-functional requirements 35

3.3.1. Architecture 35
3.3.2. Performance 36
3.3.3. Usability 36
3.3.4. Compatibility and portability 37
3.4. Summary 37

Architecture Proposal 39
4.1. Overview 39
4.2. Client 40
4.2.1. JavaScript 41
4.2.2. User interface 44
4.3. Server 45
4.3.1. Services 46
4.3.2. Data structure 47
4.3.3. Normalization 49
4.4. Real-time collaboration 50
4.5. Summary 50

Implementation 53
5.1. System description 53
5.2. User interface 56
5.2.1. Workspace 56
5.2.2. User account settings 60
5.2.3. Project management 62
5.2.4. Concept and relation annotation 64
5.2.5. Import and export documents 67
5.2.6. Automatic annotation services 69
5.2.7. Real-time collaborative features 70
5.3. Implementation details 71
5.3.1. Document parsing and representation 71
5.3.2. Import and export documents 75
5.3.3. Annotation services 75
5.3.4. Web-service security and role-based access control 77

Results and discussion 81
6.1. BioCreative IV experiment 81
6.2. Results 82

Conclusion 87
7.1. Future Work 88

Bibliography 91

List of Figures

Figure 1: Global pipeline of text mining solutions (adapted from [4]) 3

Figure 2: Medline growth over the last years. 7

Figure 3: Processing steps and respective dependencies and resources currently applied on biomedical information extraction. 10

Figure 4: Illustration of NLP tasks and their dependencies. 12

Figure 5: Illustration of the biomedical concept recognition task, where each recognized concept name have an associated unique identifier from a curated resource. 14

Figure 6: General processing pipeline of biomedical NER solutions. 17

Figure 7: Relation illustration with the sample sentence "Alpha-synuclein and parkin contribute to the assembly of ubiquitin lysine 63-linked multiubiquitin chains." 20

Figure 8: General processing pipeline of relation and event extraction solutions. ..21

Figure 9: General overview of the implementation of Egas. 40

Figure 10: General client-side architecture of Egas 43

Figure 11: Project data structure. 48

Figure 12: Meta-annotations data structure. 48

Figure 13: Document annotations data structure. 48

Figure 14: Curation time data structure. 49

Figure 15: Illustration of the technique applied by Node.js in order to enable real-time collaborative features. 50

Figure 16: Organization of Egas based on projects, users, documents and respective annotations. 54

Figure 17: Typical usage pipeline of Egas presenting the provided features. 55

Figure 18: Egas's main user interface interaction areas. 57

Figure 19: Egas's main user interface action components. 57

Figure 20: Inline concept visualization and nested concepts. 58

Figure 21: Inline relation annotations visualization with bi-directional relations.... 58

Figure 22: Project navigator, document navigator and document switcher. 59

Figure 23: Project search box with suggested results. 59

Figure 24: Visualization of advanced search window. 60

Figure 25: Account settings window. 61

Figure 26: Create new project form. After creating a new project, users can easily configure it using the project wizard. 61

Figure 27: Annotation guidelines in project administrator window. 62

Figure 28: User management in project administration window. 62

Figure 29: Management of target concepts to annotate. 63

Figure 30: Management of target relations to annotate. 63

Figure 31: Project statistics in project administration window. 63

Figure 32: User interface interaction steps to add a new concept annotation. 64

Figure 33: User interface of adding a new concept annotation and associate it to a unique identifier. 65

Figure 34: User interface interaction steps to add a new relation. 66

Figure 35: Context menu of document annotations. 66

Figure 36: Mouse over popup showing normalization information of a concept annotation. 67

Figure 37: Import documents from local source. 68

Figure 38: Import documents from remote sources by PubMed and PubMed Central identifiers. 68

Figure 39: Search and import documents from PubMed and PubMed Central. 68

Figure 40: User interface to export project documents and respective annotations. 69

Figure 41: Integrated automatic annotation services. Users only need to indicate the documents that should be annotated by the service. 69

Figure 42: Egas's real-time collaborative features, providing instant feedback when other users are annotating documents. 70

Figure 43: Egas's collaborative chat window implemented by taking advantage of TogetherJS library. 70

Figure 44: General overview of required procedures to load a document and respective annotations in Egas's workspace. 72

Figure 45: Representation of line splitting process. The top of the figure represents the *div* element containing the entire sentence. In the end of this process, as we can see in the bottom of this figure, each line is represented inside an individual *div* element. 73

Figure 46: Display of concept annotations that start and end in different lines. 73

Figure 47: Different representation of the relation description, placing the label where space is available. 74

Figure 48: Representation of a relation that begins in one line and ends in the next one. 74

Figure 49: User interface to associate the target concepts of the selected project to the ones provided by the BeCAS annotation service. 76

Figure 50: Summary of the systems that participated in the BioCreative IV IAT task, presenting the target goals and supported tasks, namely triage, entity recognition, event mining, and entity normalization [59]. Note that when we participate in this task, Egas not supported yet normalization. 83

Figure 51: Illustration of the overall satisfaction of curators per system, including user experience (blue), the rating of the system (orange), and the recommendation of the system (green) (adapted from [59]). 84

Figure 52: System performance of Egas (green) and Brat (red) when loading different articles. 85

List of Tables

Table 4.1: How HTTP methods are typically used to implement a RESTful API. 46

Table 5.1: Different roles and respective permissions in Egas user interface............78

Chapter 1

Introduction

Growing availability of data creates wide-ranging opportunities and challenges. A recent study, conducted by Hilbert and López [1] shows that the capacity of the World to store and exchange information in the last 20 years have grown at a rate of at least 23% a year. In 1986, we had the capacity to store something like 540 MB per person, and almost 20 years later, in 2007, more that 400 billions CDs were required to store all the information, almost 43 GB per person. Thus, the need to analyze and structure all this data is imminent. With the exponential growth of the available information, the efficiency and speed that we take to transform this data into knowledge turns out to be a differentiating factor to beat this challenge.

As of 2007, 94% of all the technological memory was presented to us in digital format, nevertheless, in the year of 2000, 75% of all information was still in analog format. On the other hand, nowadays, most of the public data is available in digital formats, but “this is nothing more than a blink of an eye in historical perspectives" [1]. This data availability facilitates the access by computerized solutions to the information, however, the lack of structured information is making it hard to analyze and extract knowledge from it.

The term "knowledge" is sometimes viewed in a hierarchical sense, with data as the base, then additional narrowing layers adding information, knowledge, understanding and wisdom [2]. The normal workflow to access information is the search for the specific data, extract information from it and understand the meaning of it to create wisdom. This type of data is also known as unstructured data, since it is not organized and does not follow a specific data model. Merrill Lynch estimates that more than 85 percent of all business information exists as unstructured data – commonly appearing in e-mails, memos, notes, news, chats, reports, letters, marketing material, research, presentations and Web pages [3].

Therefore, this explosion of available data and information creates a wide range of opportunities that require huge amounts of expensive resources, human or technical, which demanded the creation of tools that effectively manage unstructured data in order to obtain the desired outcomes. Going further, by associating the information extracted from large amounts of data, computerized solutions may also contribute to discover hidden relations that enable the discovery of new knowledge.

With this constant increase of information recorded in texts, there is high research interest in techniques that can identify, extract, manage, integrate, and exploit it. Text Mining (TM) is the field of Data Mining (DM) that deals with those requirements, by deriving high-quality information from text. The primary goal of text mining is to retrieve information that is hidden in text, presenting it in a concise and simple form to final users to enable the extraction of knowledge [4].

In order to achieve this goal, two main subject areas were defined:

- Information Extraction (IE): extract specific information from unstructured data, building a structured and unambiguous representation of chunks of text and relations between them;
- Information Retrieval (IR): representation, storage, organization and access to information items, providing easy access to the information in which the final user is interested.

In a Text Mining system, the main input comes, almost every time, from natural language texts. These texts are processed by information extraction procedures to extract specific information in a structured way, enabling its comprehension by computers. In the end, the information from unstructured data is filtered and presented in a simple and structured form, focusing on the information requested by final users. Figure 1 presents the global pipeline of a Text Mining system, presenting the results provided by each task and the relations between them.

IE and its several methods were introduced by the Message Understanding Conferences (MUCs), which defined the requirements, evaluation strategies and the several tasks that need to be performed in order to accomplish the IE idea and goals successfully:

- Named Entity Recognition (NER): identify atomic elements in text as specific entity names, such as people and organizations;
- Normalization and disambiguation: associate an unique meaning to a concept name (e.g., “June” could refer to a person’s name, a calendar month or a gene);
- Coreference: identify when two different expressions refer to the same concept (e.g., “Tom” and “he” in the same sentence may refer to the same person);

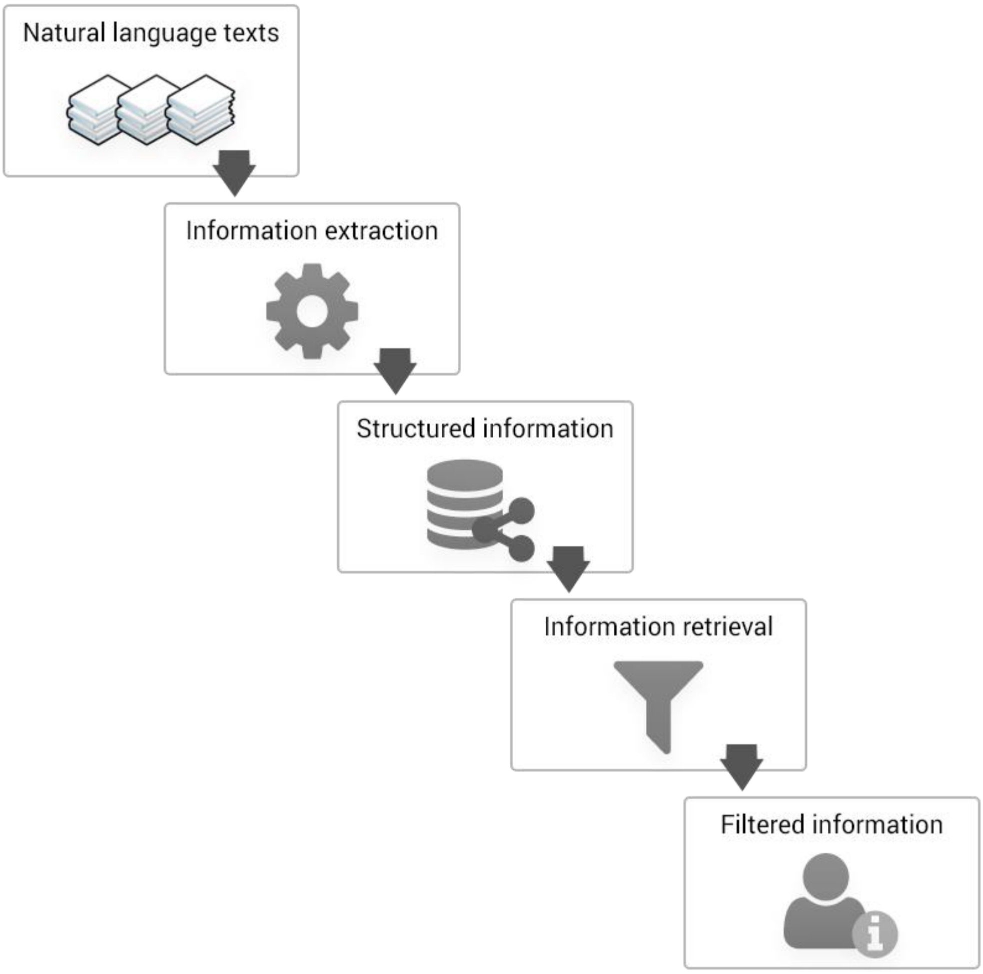


Figure 1: Global pipeline of text mining solutions (adapted from [4])

- Relation mining: extract relations between concepts (e.g., considering the entities “Europe” and “Portugal”, the relation “Portugal -> Country -> Europe” should be extracted if it is present in the text in some manner);
- Summarization: extract and compile main ideas of a text based on a specific goal;
- Classification: identify prime themes of a specific text (e.g., sports, politics, and arts).

In the seven editions of MUC, they applied the previously described tasks on people and organizations information mining, defining state-of-the-art solutions and baseline results for IE.

As we can see, automatic extraction of biomedical information is very challenging and with many related issues, because of the complexity of the biomedical field and the many ambiguities that may be found on scientific documents.

On the other hand, in order to provide trustful and valid biomedical knowledge bases, manual annotation of scientific articles is also performed by expert biocurators. A biocurator is someone that curates, collects, annotates, and validates information that is disseminated by biological and model organism databases [5]. The biocurator's work involves quality control of primary biological publication, extracting and organizing data from scientific literature, and describing the data width standard annotation protocols. Recognized as the "museum catalogers of the Internet age" [6, 7], biocurators enable the discovery of new knowledge and biological database inter-operability. Likewise, the primary goals of biocuration are based on accurate and comprehensive representation of biological knowledge, as well as providing easy access to these data for working researchers. Such goals can only be achieved thanks to the efforts of biocurators, but also software developers and bioinformaticians.

Although, when updating the knowledge bases or generating annotated resources, mistakes cannot be forgotten. Therefore, the automatic information must be carefully analyzed and corrected. According to this, many studies have shown that the curation time can be improved by using automatic solutions to assist the expert curators [8, 9]. However, these solutions are not being widely used by their main target audience, the broad biological communities [10]. This is easily proved, if we take into consideration the experiment organized by Hirschman et al. [11], which included 30 biocurators from 23 different databases. They concluded that two-thirds of the participants had already experimented text mining, but less than half were using it in some aspect of curation, mainly due not only to the lack of standards and interaction between biocurators and developers, but also because of the complexity and ambiguity of biocuration tasks.

Bolchini et al. [12], also proved that users simply do not use these resources and tools if they were not able to find all the information needed in their daily research activities, showing that usability on bioinformatics tools is extremely important and relevant. The interface with the professional curator is fundamental since it influences the adaptation to such resources. In order to achieve that aspect, the development of interactive solutions that take advantages of automatic systems is urgent, once it can easily and more effectively help biocurators to keep current knowledge bases updated, and generate annotated data to develop and evaluate automatic solutions.

1.1. Motivation

Manual annotation of large amounts of biomedical literature can be a very demanding and exhausting task. In order to organize and manage these data, several manual curation efforts have been set up to store the extracted information in structured resources. However, the associated costs can make it very expensive and difficult to keep these databases up-to-date.

These factors, led to an increasing interest of Text Mining techniques to perform automatic information extraction from scientific documents. Nevertheless, in fields such as biomedicine, with high levels of complexity, automatic information extraction remains very challenging, and even the most advanced solutions lack accuracy and professional user interfaces.

Moreover, due to this gap, solutions are still not widely adopted by biomedical research communities. As presented before, the usability of bioinformatics resources is fundamental to effectively support users in their daily research activities [12]. Thus, our motivation rely on the development of an interactive solution that take advantage of automatic algorithms and existing knowledge resources to assist expert curators in their daily tasks. The application, its interface and the available features, must be carefully analyzed and designed in order to provide curators a state-of-the-art solution in interactive mining.

1.2. Results

The work developed in this thesis generated a web-based tool for biomedical text mining and document annotation, as well as two publications:

- Egas, a web-based platform for biomedical text mining and collaborative curation, http://bioinformatics.ua.pt/egas/
- D. Campos, J. Lourenço, T. Nunes, R. Vitorino, P. Domingues, S. Matos, and J. L. Oliveira, *Egas - Collaborative Biomedical Annotation as a Service*, in Fourth BioCreative Challenge Evaluation Workshop, Bethesda, Maryland, USA, Oct. 2013, p. 254–259;
- D. Campos, J. Lourenco, S. Matos, and J.L. Oliveira, *Egas: a collaborative and interactive document curation platform*. Database (Oxford), June 11, 2014.

1.3. Thesis outline

The remaining chapters of this thesis are organized according to the following:

Chapter 2 presents a detailed analysis of the state-of-the-art in the biomedical information extraction domain. Firstly, we describe the pre-processing tasks and techniques that enable the automatic application of information extraction solutions. Then we will focus on concept recognition, relation mining and finally on interactive mining and data curation.

Chapter 3 will list the user requirements that shaped the development of our solution, and explore the limitations and issues that need to be covered in order to deliver a high-level solution to biomedical data curation. Thus, functional and non-functional requirements of the desired solution will be analyzed in detail and presented in this chapter.

Chapter 4 presents Egas architecture. Firstly, we describe the overall architecture and then we provide more detailed information regarding the client-side and the server-side of the application. Data structures as well of some aspects related to the collaborative features and the normalization functionalities are also provided.

Chapter 5 presents the overall implementation of the designed solution. Starting with a system description and general exposition of the application workflow, we provide detailed information regarding to system implementation and the inherent challenges. Then, we present Egas' user interface, providing functionality details and the main implementation issues. Finally, we present some implementation algorithms and features that required more ingenious techniques, such as document parsing and representation, annotation services and import and export.

Chapter 6 presents the results from our participation in the BioCreative IV Interactive Annotation Task experiment.

Chapter 7 presents the concluding remarks of this thesis and highlights directions for future work in order to improve to developed solution.

Chapter 2

Biomedical information extraction

Nowadays, with the exponential growth of biomedical textual information that is produced, it is extremely important that we efficiently analyze and structure these data to extract knowledge to structured databases. Information extraction and knowledge discovery have been attracting a giant amount of research, as well as industry and media attention. Across a wide variety of fields data are being collected and accumulated at a dramatic rhythm.

Over the last 20 years, the total size of the Medical Literature Analysis and Retrieval System Online (MEDLINE) database, has been growing at a ~4,2% annual rate. As of 2005, there were more than 15 million publications in MEDLINE [13]. Since then, approximately 2000 to 4000 new entries have been added to the database every day, exceeding 20 million in 2012 (Figure 2).

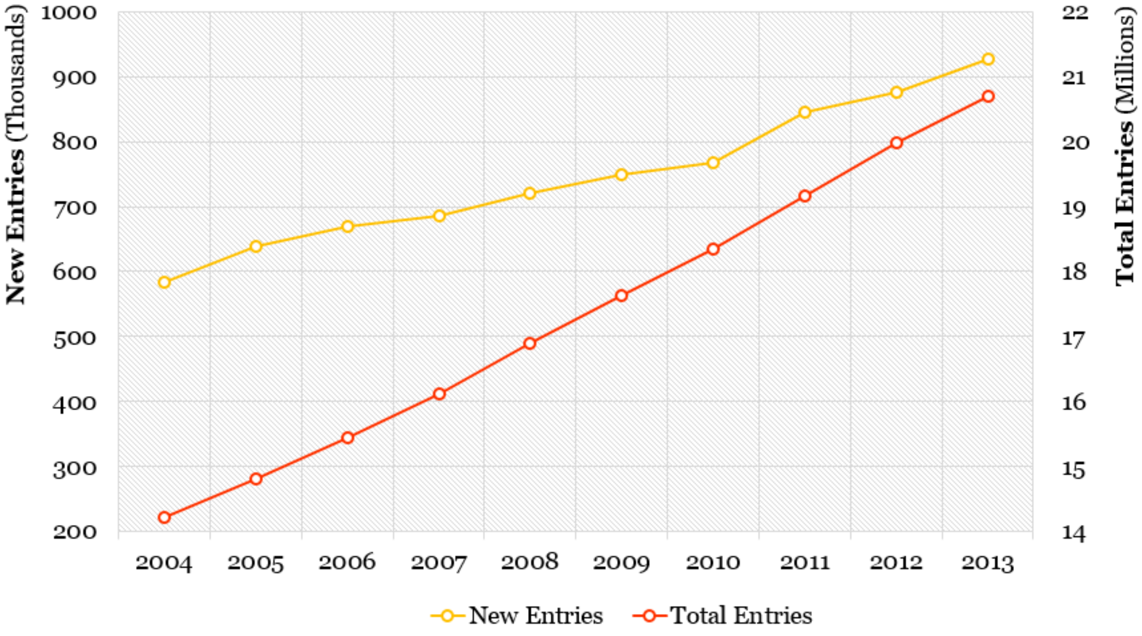


Figure 2: Medline growth over the last years.
(adapted from http://www.nlm.nih.gov/bsd/bsd_key.html)

Many of biomedical resources such as MEDLINE started to manually curate these scientific articles to maintain their existing knowledge resources updated. However, with the huge amounts of data produced every day, this task became significantly harder and expensive to do. These factors have naturally led to an increasing interest in the development of computerized solutions to extract specific biomedical information from scientific articles to perform those tasks automatically and keep the databases updated [14].

Some scientific domains, like biomedicine, reveal diverse complex challenges that difficult the application and development of text mining solutions such as the wide range of interrelated concepts that biomedical knowledge covers. However, it is very important to connect both ends of this field in order to know "how" and "why" things happen.

In 1989, Russell L. Ackoff said that "information is data that has been given meaning by way of relational connection. This "meaning" can be useful, but does not have to be. In computer parlance, a relational database makes information from the data stored within it." [15].

Therefore, biomedical domain is divided in many fields and sub-fields, with very restrict communication and it is very hard to link concepts between then. On the other hand, the biomedical field is in constant evolution and new concepts, knowledge and theories are emerging almost every day. Likewise, the specialized non-standardized terminology, results in high levels of ambiguity between articles and the same terms are constantly represented by slightly different terminologies. This way, the development of text mining solutions in this area is considered a proving ground for the application and development of innovative information extraction solutions, since it is assumed that a technique that performs well in the biomedical domain will perform equally well in a different and simpler domain [16].

On the other hand, the current automatic systems are less powerful than expert curators, since it is very difficult to focus all curator's domain knowledge into structured representation. Nonetheless, even different expert annotators have different interpretations of the same data, which results in active and complex discussions with different opinions. These differences may lead to inconsistencies in the final annotated data used to train text mining solutions, which may affect the precision and quality of their results.

Nevertheless, collecting biomedical information from scientific articles is extremely important and contributes, every day, for keeping the knowledge bases updated and generates new hypothesis for knowledge discovery. Therefore, regarding the applicability of TM systems in biomedical real-life problems, there are various examples, specifically in pharmacovigilance, drug discovery and drug repurposing. The beneficial effects of fish oil to patients with Raynaud's disease, and the potential of magnesium to treat migraines, were the first scientific hypothesis discovered by text mining solutions. They were presented by Swanson, in 1990, and both connections were validated in clinical trials and became well established in nowadays clinical practices [17]. On the other hand, the EU-ADR European project [18] developed an innovative computerized system [19] for pharmacovigilance, detecting adverse drug reactions to supplement spontaneous reporting systems and translating scientific and clinical evidences into patient safety and health benefit. The authors applied text mining techniques to analyze electronic health records of 30 million patients in order to detect "signals", i.e., combinations of drugs and suspected adverse events that warrant further investigation. With this system, the authors confirmed the association between the use of non-steroidal anti-inflammatory drugs and upper gastrointestinal bleeding.

These examples show us the success of applying text mining solutions on real-life problems, which provide among another benefits, the need for less money and time for drug discovery, testing and vigilance, which results in better and improved healthcare services.

The solution presented in this thesis is focused on three essential biomedical information extraction tasks, namely concept recognition, relation extraction and interactive mining. In the next pages a detailed description of these tasks is provided, as well as their goals and associated challenges, applied approaches, existing solutions and performance results of IE solutions. These analyses provide a description of current state-of-the-art work in biomedical IE, defining the platform for further improvements and research.

Figure 3 presents the processing steps considered in this analysis:

- Pre-processing: processes the input documents to simplify IE tasks;
- Evaluation: established metrics to understand the behavior of IE systems and compare different approaches;
- Concept recognition: identify concept names and associate unique identifiers from knowledge-bases;
    - Named entity recognition: identify concept names;
    - Normalization and disambiguation: associate unique identifiers to previously recognized names;
- Relation mining: extraction of relations between previously annotated concepts;
- Interactive mining: curation of biomedical texts with both manual and automatic systems.

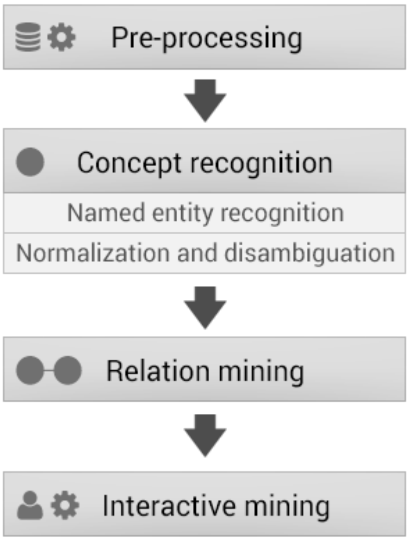


Figure 3: Processing steps and respective dependencies and resources currently applied on biomedical information extraction.

2.1. Resources

Resources are used to support the development of biomedical information extraction systems, providing input data that allow the development of automatic solutions and platforms to evaluate and compare different approaches.

One of the most important resources of biomedical information extraction solutions are knowledge bases, namely databases and ontologies. Despite the fact that ontologies provide an excellent way to represent reality, databases still are the better method for storing and searching data when this is of considerable size [20]. Most of these knowledge bases are focused on gathering detailed information regarding specific concepts, namely genes and proteins, drugs, chemicals and species. For example, Uniprot is a database that aims to provide a central resource for protein sequences and functional information [21]. Furthermore, Gene Ontology (GO) [22] provides a set of structured vocabularies for specific biological domains.

Considering the complexity of the biomedical domain, researchers started working on techniques to integrate different knowledge bases. The Resource Description Framework (RDF) [23] is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. By taking advantage of the RDF specification, researchers are able to integrate heterogeneous resources in a unique resource maintaining existing links between concepts from different knowledge bases [24]. The European Bioinformatics Institute (EBI) also developed an RDF platform to enable easy access and integration of gene expression data [25].

On the other hand, another important resource of biomedical information extraction systems are corpora. A corpus is a large and structured set of texts that usually contain annotations regarding specific domains and/or tasks, which are used in the development and evaluation of implemented solutions. The information provided in the corpus is extremely important, since the development of information extraction solutions are highly dependent on it.

The two following existing types of corpora, vary with the source of the annotations:

- Gold Standard Corpus (GSC): annotations are manually created by expert curators, following specific and detailed guidelines;
- Silver Standard Corpus (SSC): annotations are automatically generated by computerized solutions.

The quality and agreement of manually annotated corpora may be evaluated by metrics such as the Inter Annotator Agreement (IAA). Therefore, a low IAA reflects the disparities between the annotations performed by the human experts that provide inconsistent information that hampers the development and evaluation. Usually only small amounts of documents are provided, because of the efforts and associated costs required to build such corpora. However, the automatically generated corpora are highly useful, since it provides a huge amount of information. Nevertheless, this information might not be trusted because of the large amount of present mistakes. Corpora also vary in the granularity of provided texts, considering full-text documents, just theirs abstracts or selected sentences. Tasks that do not require information context are usually targeted by sentence-based corpora, however the complete paragraph, section or document might be required. Full-text documents typically hold more information than their abstracts, but are computationally less efficient to process. It is also important to record that Schuemie et al. concluded that the results section, on a document, is the one that provides the highest information coverage [26].

Therefore, knowledge bases and corpora are extremely important to develop and evaluate information extraction solutions.

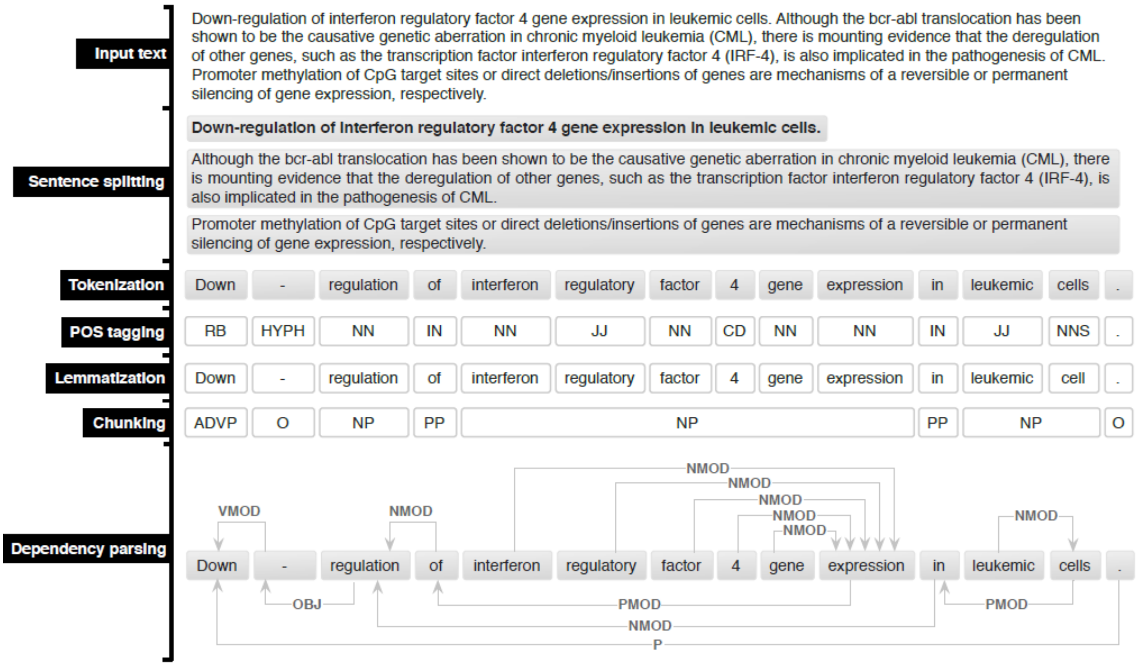
2.2. Pre-processing

In order to perform biomedical information extraction, various small tasks of pre-processing are usually applied, namely tokenization, stopword removal and natural language processing.

Natural Language Processing (NLP) solutions have been researched and developed for many years. It was first described in 1960 as a sub-field of Artificial Intelligence and Linguistics, with the goal of studying problems in the automatic generation and understanding of natural language. Nowadays, these solutions can be effectively accomplished by computerized systems, however, in information extraction systems, first we need to properly delimit the documents into meaningful units, like sentences. Therefore, most NLP solutions split each sentence into tokens, which are the basic units of data processing. Thus, and since real-world documents lack such structure information, it is necessary to perform various pre-processing methods before information extraction tasks.

On the other hand, and due to the specificity of biomedical domain, common English solutions does not provide the best results when applied to scientific documents. He and Kayaalp [27] analyzed the application of several tokenizers on biomedical texts and concluded that most solutions are too simple for biomedical information extraction, showing that domain specification is fundamental to uplift the performance of the system.

In Figure 4 some linguistic processing tasks and their dependencies are presented, illustrating the provided output considering the sentence "Down-regulation of interferon regulatory factor 4 gene expression in leukemic cells". As we can see in the dependencies between tasks, POS tagging should not be performed before tokenization, since the tokens are essential to assign the linguistic role tags.



Down-regulation of interferon regulatory factor 4 gene expression in leukemic cells. Although the bcr-abl translocation has been shown to be the causative genetic aberration in chronic myeloid leukemia (CML), there is mounting evidence that the deregulation of other genes, such as the transcription factor interferon regulatory factor 4 (IRF-4), is also implicated in the pathogenesis of CML. Promoter methylation of CpG target sites or direct deletions/insertions of genes are mechanisms of a reversible or permanent silencing of gene expression, respectively.

Down-regulation of interferon regulatory factor 4 gene expression in leukemic cells.

Although the bcr-abl translocation has been shown to be the causative genetic aberration in chronic myeloid leukemia (CML), there is mounting evidence that the deregulation of other genes, such as the transcription factor interferon regulatory factor 4 (IRF-4), is also implicated in the pathogenesis of CML.

Promoter methylation of CpG target sites or direct deletions/insertions of genes are mechanisms of a reversible or permanent silencing of gene expression, respectively.

Input text
Sentence splitting
Tokenization
Down - regulation of interferon regulatory factor 4 gene expression in leukemic cells .
POS tagging
RB HYPH NN IN NN JJ NN CD NN NN IN JJ NNS .
Lemmatization
Down - regulation of interferon regulatory factor 4 gene expression in leukemic cell .
Chunking
ADVP O NP PP NP PP NP O
Dependency parsing
Down - regulation of interferon regulatory factor 4 gene expression in leukemic cells .
VMOD
NMOD
NMOD
NMOD
NMOD
NMOD
NMOD
NMOD
OBJ
PMOD
NMOD
P
PMOD

Figure 4: Illustration of NLP tasks and their dependencies.

Sentence splitting is used to break the text document into its respective sentences. Several solutions were developed to perform this task on biomedical texts, namely

Lingpipe[1], GENIA SS [28], JSBD [29], OpenNL[2] and SPECIALIST NLP[3]. Some solutions can achieve an F-measure of 99%.

The next step is tokenization, which is the process of breaking the previous split sentences into constituent meaningful tokens. Since all the following steps will be based on tokens, this is one of the most important tasks of pre-processing in the information extraction workflow. Various systems were developed for the biomedical domain, such as GENIA Tagger [30], JTBD [29], and SPECIALIST NLP. Some of them presented performance results achieving 96% of F-measure.

Lemmatization is used to group together inflected forms of a word in order to interpret them as a single token. Therefore, lemmatization is an important step that finds the origin of each word, for instance, the lemma of “writing” is “write”. GENIA Tagger and BioLemmatizer [31] are examples of solutions for biomedical lemmatization. The best performance results achieve an F-measure of 97%.

POS tagging is the process of associating tokens with a particular grammatical category based on its definition and context, in order to understand the linguistic role of each word – token – in a sentence. Likewise, each token is tagged, for instance, as a Noun (NN), an Adjective (JJ) or an Adverb (RB). Solutions like GENIA Tagger, Lingpipe and OpenNLP are examples that support biomedical POS tagging. In this task, some systems can achieve an F-measure of 90%.

However we still need to understand the structure of a sentence. Likewise, chunking splits a sentence into groups of tokens that compose a grammatical unit, like noun phrase (NP), verb phrase (VP) and preposition phrase (PP). GENIA Tagger, Lingpipe and OpenNLP are also systems that provides this task. The best performing solutions can achieve 95% of F-measure.

On the other hand, it is very important to understand in detail how tokens and chunk phrases are related in the sentence, providing an syntactic analysis and ultimately the meaning of the sentence. Dependency parsing identifies the relations beetween each chunk phrases and categorizes them according to its grammatical roles, such as noun modifier (NMOD), verb modifier (VMOD) and preposition modifier (PMOD). Various solutions support biomedical dependency parsing, such as GDep [32] and Enju [33]. Other more generic parsers have also been applied to biomedical literature such as Berkeley[4] [34] and Stanford Parser[5] [35, 36]. In the biomedical domain, the best solutions achieve a top F-measure performance of 65% [37].

[1] http://alias-i.com/lingpipe

[2] http://opennlp.apache.org

[3] http://lexsrv3.nlm.nih.gov/Specialist

[4] https://code.google.com/p/berkeleyparser/

[5] http://nlp.stanford.edu/software/eventparser.shtml

Finally, in biomedical domain, to improve performance result and reduce the size of data to be processed, one of the most commonly used techniques is to discard words that we already know that are noninformative – stopwords -, like “as”, “due”, “just” and “than”. PubMed[6] provides a list of these words that can be applied in biomedical information extraction solutions.

2.3. Concept recognition

A concept corresponds to a biomedical entity that may be found on an annotated and/or specialized resource, used to represent and map knowledge. On the other hand, a resource is a database and/or ontology that contains and relates information regarding a specific knowledge sub-field, where each concept has a unique identifier. For instance, “BRCA1” is a protein that is present on the Uniprot database as a concept with the unique identifier “P38398”. Likewise, concept recognition allows us to automatically extract names of concepts and relate them with unique identifiers from curated resources (Figure 5). Applying this technique, and considering dozens of concept types, it is automatically possible to extract names of several biomedical concepts from millions of information instances.

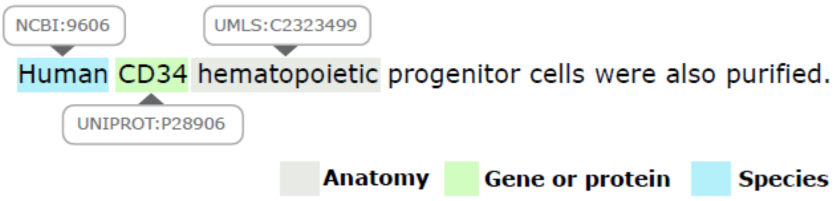


Figure 5: Illustration of the biomedical concept recognition task, where each recognized concept name have an associated unique identifier from a curated resource.

Concept recognition is a key phase in information extraction, since the success of all the next phases depend on it. Although, biomedical documents brings forward various challenges that make the application of these techniques even harder. The main challenge is related with terminology, due to the complexity of the used terms for biomedical concepts and processes [38, 39]:

- Non-standardized naming convention: a concept name may be found in various spelling forms (e.g., “N-acetylcysteine”, “N-acetyl-cysteine”, and “NAcetylCysteine”);
- Ambiguous names: Depending on the text context, a name could be related with more than one concept;

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

- Abbreviations: abbreviations are often used (e.g., “TCF” may refer to “T cell factor” or to “Tissue Culture Fluid”);
- Descriptive naming convention: lots of concept names are descriptive, which become the task of recognize them very complex (e.g., “normal thymic epithelial cells”);
- Conjunction and disjunction: two or more concept names sharing one head noun (e.g., “91 and 84 kDa proteins” refers to “91 kDa protein” and “84 kDa protein”);
- Nested names: a name could be found within a longer name, as well as independently (e.g., “T cell” is nested within “nuclear factor of activated T cells family protein”);
- Names of newly discovered concepts: there is an overwhelming growth rate and constant discovery of novel biomedical concepts, which takes time to register in curated nomenclatures.

In order to build a rich and reliable information profile, is important to automatically extract, biomedical concepts from the whole spectrum of biomedical knowledge. Typically, the following ones are the most interesting, given their implications and inherent interactions and relations:

- Species or Organism
- Gene or protein
- Enzyme
- Mutation
- Drug
- Chemical
- Anatomy
- Disorder
- Pathway
- Biological process
- Molecular function

2.3.1. Resources

Knowledge bases

There are several regulatory agencies that created standards for concept name definition in order to create unique and centralized resources and stimulate their connection with patient health records and research laboratory resources. Nevertheless, even with the success of most of these standardization processes, several main concepts still need standards for meticulous names definition. Moreover, Tsuruoka et al [40], concluded that there is an average of 5 to 14 different names of each identifier in gene and protein databases, which represents the need to gather this information on a single resource.

Corpora

Although we find relevant corpora for a wide range of biomedical concepts, most of the research efforts have been on the recognition of gene and protein names. Such effort is a consequence of different factors, namely the importance of genes and proteins on the biomedical field, the high variability of names and the urge of standardization. On the other hand, as expected, there is a significant difference in the amount of provided information between silver and gold standard corpora. This difference, caused by the necessity of using specialized human resources to create gold standard corpora is also observed when comparing corpora with and without information on identifiers, where typically, corpora without identifiers provide a higher amount of sentences. There is a clear recent trend on corpora with full-text articles and heterogenous concept types, reflecting the progress on this field, with powerful solutions capable of processing large amounts of documents and annotating multiple concept types.

2.3.2. Named entity recognition

In order to recognize concept names and associate them to unique identifiers, biomedical concept recognition can be divided into two different steps, named entity recognition and normalization and disambiguation.

Named entity recognition aims to identify chunks of text and associate them with their specific concept type. This task can be performed by different approaches, such as dictionary matching, rule based or machine learning solutions. The processing workflow of these systems generally integrates common steps and resources presented in Figure 6:

- Corpus: groups of related text documents;
- Pre-processing: tasks performed to simplify and enable named entity recognition process;
- Named entity recognition: automatically recognize concept names;
- Normalization and disambiguation: associate the correct unique identifiers to the previously recognized concept names;
- Post-processing: tasks related to refinement of recognized concept names;
- Annotated corpus: documents containing recognized concept names.

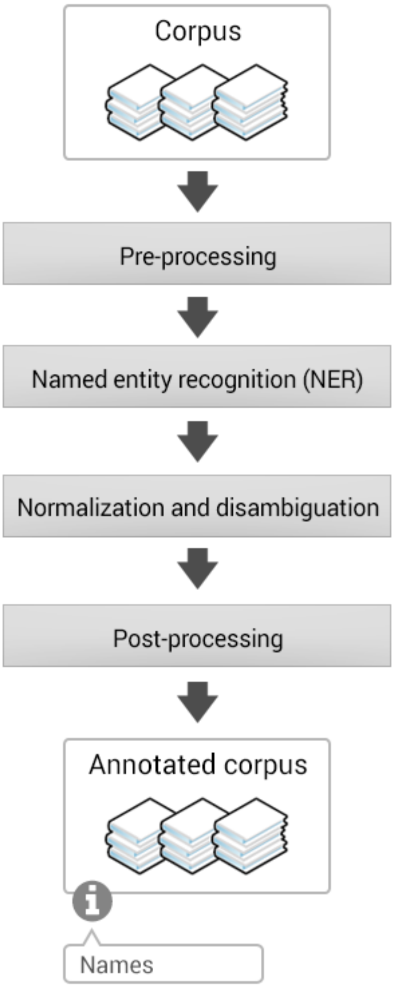

Corpus
Pre-processing
Named entity recognition (NER)
Normalization and disambiguation
Post-processing
Annotated corpus
Names

Figure 6: General processing pipeline of biomedical NER solutions.

Different approaches may be similar in terms of processing pipeline, however, each one can be more appropriated to fulfill different requirements, depending on the concepts that we need to identify. Likewise, it is recommended to use different approaches according to the requirements of each concept type:

- Rule-based: concepts with a strongly defined orthographic and morphological structure;
- Dictionary-based: closely defined vocabulary of names;
- Machine learning based: strong variability and dynamic vocabulary of names.

However, since each approach require different implementations, sometimes is not possible to apply the best approaches.

2.3.3. Normalization and disambiguation

The second step of biomedical concept recognition is normalization and disambiguation. The goal of normalization is to associate each identified chunk of text with a unique concept from a curated knowledge base. Such process is performed by associating unique concept identifiers from databases and/or ontologies with each chunk of text previously recognized.

The techniques applied in this process are similar to the methods applied on dictionary-based approaches for NER. However, the matches are performed between the dictionary's entries and the chunks of text previously recognized as entity names, which allows performing a more flexible matching through approximate matching approaches or regular expressions.

This task starts by associating the recognized name with any name on biomedical resources. If there is no associated identifier, there is no option to assign an identifier to this concept, so it may be discarded as an entity name. Likewise, if there is only one identifier associated, it is immediately assigned. On the other hand, the entity name can be associated with multiple identifiers, in this case, it is considered ambiguous. Due to the complexity and extensibility, in biomedical domain, ambiguity is very usual. For instance, Jimeno-Yepes and Aronson [41] conclude that the most ambiguous term in MEDLINE is “study”, which is associated to six different concepts and occurs more than three million times.

On the other hand, the second step of this task consists in disambiguation. This process aims to associate ambiguous names to the correct concepts. When this process is successfully performed it increases the number of biomedical concepts normalized correctly, improving concept recognition and overall information extraction. Thus, Word Sense Disambiguation (WSD) aims to develop solutions to minimize this problem by identifying the meaning of ambiguous terms in a specific context [42, 43].

To improve concept disambiguation, several corpora were specially built for this purpose, providing terms, associated meanings and text passages for each meaning. There are various public corpora available to support the development and evaluation of WSD solutions, such as NLM WSD [44], with more than 30 thousand annotated MEDLINE abstracts and 203 ambiguous entities with almost 38 thousand occurrences, Medstract [45] directed to acronym disambiguation and MuchMore [46] containing both English and German versions of more than 7000 abstracts.

2.4. Relation Mining

To provide a better interpretation of biological processes, such as gene transcription, protein binding or cell cycle regulation, it is very important to identify biomolecular events. These events and their biological significance and impact, are commonly described in scientific literature and identifying their complex chains and networks is a very challenging and time-consuming task. However, this important knowledge can also be used by other industries, such as pharmaceutical to improve drug discovery solutions, since the identification of proteins involved in key events might result in the subsequent uncovering of new drug targets [47]. Likewise, by processing millions of scientific articles and through the application of the ABC model defined by Swanson [48], the automatic extraction of relations between concepts may contribute to new findings, generating new knowledge [17], since it helps to find hidden biological relationships. Therefore, relation and event mining is an extremely important and well established way to extract information from biomedical documents [49], gaining everyday more attention by research groups around the globe.

Due to its importance, first research solutions were focused on extracting direct associations between two concepts, which are known as binary relations. Figure 7 demonstrates a textual representation of binary relations. On the other hand, considering the important information obtained by identifying such relations, it has been applied on different tasks and targeting a wide range of domains, such as [50-52]:

- Protein-Protein interactions: contribute to a better understanding of biological functions and molecular processes;
- Gene-Drug: understand how specific drugs can be tailored to specific genetic contexts;
- Gene-Disorder: understand the role that genetic information plays on specific diseases and/or phenotypic phenomena;
- Drug-Drug interactions: improve multi-drug therapy by understanding how one drug affects the activity of other;
- Drug-Disorder: understand how specific drugs affect specific disorders, namely adverse drug reactions to improve pharmacovigilance;
- Location: physical location associated with specific concepts, such as “contained in” and “has location”;
- Lunctional: general functional relation between concepts, such as “is caused by” and “is treatment for”.

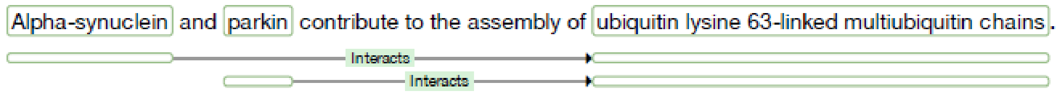

Alpha-synuclein
and
parkin
contribute to the assembly of
ubiquitin lysine 63-linked multiubiquitin chains
Interacts
Interacts

Figure 7: Relation illustration with the sample sentence "Alpha-synuclein and parkin contribute to the assembly of ubiquitin lysine 63-linked multiubiquitin chains."

Extracting binary relations allows us to collect and relate facts not possible before, however, sometimes they cannot fully represent the biological meanings of its original text, since some information can only be expressed in higher-order relationships [53]. Likewise, there was a need to develop solutions that can identify and extract complex relationships. The first biomedical event extraction tasks focused on identifying complex and nested relations, was introduced by BioNLP shared tasks [54-56]. Thus, to improve the understanding of the extracted information, not only relations between concepts were considered, but also relations between concepts and other relations and events between relations. The representation of these relations includes the two concepts and/or relations and what we call the trigger, which is a word, generally a verb or nominalized verb or adjective (e.g. "contribute", "promoting") that represents the type of relation.

In recent editions of BioNLP shared tasks (2011 [55] and 2013 [56]), new annotation tasks targeting different domains were introduced. Due to these community challenges, biomedical text mining researchers usually refer to the task of extracting direct relationships as relation mining and to extracting chains of relations as event mining. Even though event and relation mining solutions require different approaches they follow similar processing pipelines, which are composed by the following resources/steps (Figure 8):

- Corpus: annotated examples for development and/or evaluation;
- Pre-processing: processing methods to enable automatic relation mining;
- Concept recognition: automatically recognize concept names and associate identifiers from known knowledge bases;
- Document classification: in some cases, it may be useful to automatically classify the document as of interest for the target relation or not;
- Trigger recognition: identify the chunk of text that triggers the relation and serves as predicate;
- Relation extraction: automatically extract relations between concepts;
- Post-processing: refine recognized relations;
- Annotated corpus: input documents containing recognized concepts and target relations.

As mentioned before, concept recognition task has a key role in biomedical relation and event extraction, since this step relies on the success of previous steps. These tasks typically start with Gold Standard set of concepts annotated, in order to assure the flexibility of these systems to different domains. For instance, considering the sentence “BAG1 interacts with Tau.”, the protein names “BAG1” and “Tau” are converted into representative and sequential tokens, such as “PRO1 interacts with PRO2”.

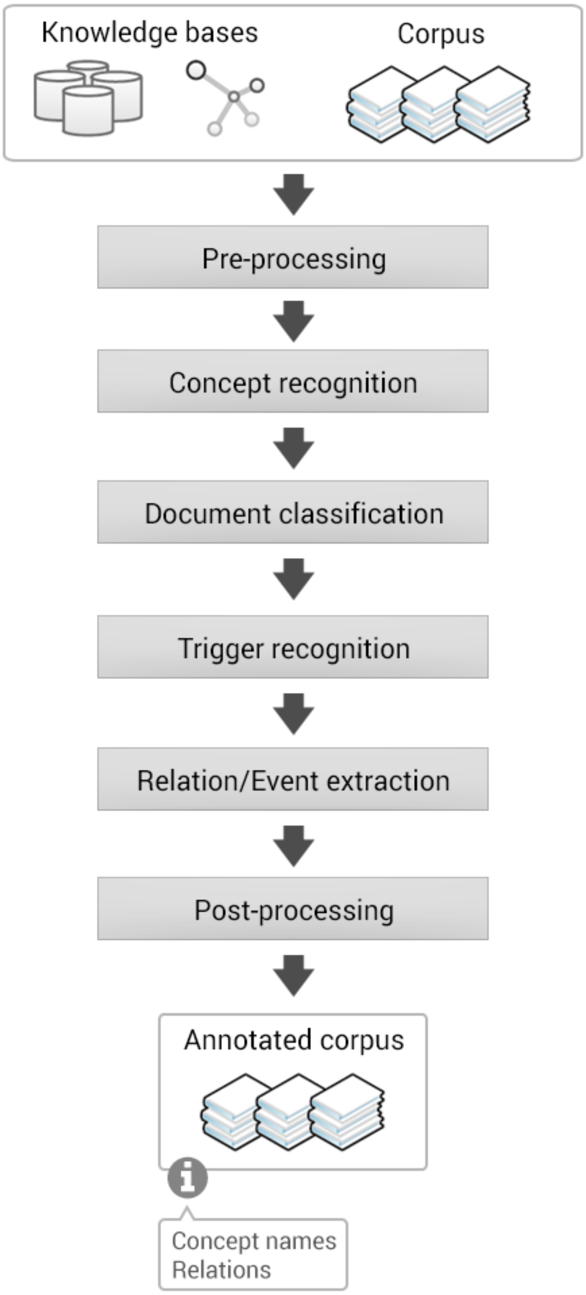


Figure 8: General processing pipeline of relation and event extraction solutions.

2.4.1. Resources

Knowledge bases

There are several relevant knowledge bases for relation mining, which are specially focused on direct associations between concepts. Protein-protein interactions reveals one of the most important fields, there are several databases covering this subject. On the other hand, there are also databases that explore drug-drug interactions and a few more that cover relations between genes, drugs and diseases.

Databases provided for binary relation mining can also be used to perform complex relation mining, however they are strongly focused on storing direct relationship between concepts. Therefore, there are various knowledge bases specifically directed to understanding biological metabolisms and extraction of events and complex relations.

Corpora

There are several corpora available to support the development of relation and event mining systems providing carefully annotated concept names and relations between them. The high number of corpora available for PPI mining reveals its importance in the biomedical domain, since it is one of the most relevant tasks of molecular biology. However, solutions of drug-drug interactions, target-disease and gene-disease relations are becoming relevant to several research groups.

Since the expensive resources, human or technical, and the effort required to manually curate these specific relations and events, the corpora size and granularity is not considerably high. Likewise, comparing these resources with the ones provided for concept recognition tasks, we must assume that these corpora are considerably small. Moreover, some researchers have already tried to perform event mining in full-text documents, concluding that different challenges were presented, resulting in lower performance results.

2.4.2. Document classification

In other to obtain the best resources to develop relation mining solutions, it is very important to select a set of documents that probably contain relations of a certain domain, such as protein-protein, gene-disease or drug-drug interactions. On the other hand, with the rapidly growing of scientific data available to curate, researchers have a hard time finding the most relevant publications to select. Therefore, since keywords are not sufficient to identify relevant articles for complex biomedical event mining [57],there are some automatic solutions developed to select articles that may be relevant for the target relation mining task.

Document classification aims to assign a score to each document, according to the probability of containing relations of a certain field. Likewise, after this task is performed, documents above a pre-defined threshold are used for processing and the remaining ones are discarded. These solutions, integrated with relation and event mining pipelines, provide two main advantages:

- Improving performance, since a large amount of not relevant documents are not processed;
- Improving processing speed by processing only the relevant documents.

2.4.3. Trigger recognition

Trigger recognition is a crucial step to successfully perform relation mining, since many approaches depend on its output to properly extract relations and events from the text. As mentioned before, events are defined around the trigger, which defines the type of event. Likewise, trigger recognition reveals a fundamental step in event mining systems. On the other hand, since binary relation corpora usually do not provide trigger annotations, it is important to create auxiliary techniques to identify triggers in order to facilitate the extraction of relations. Notwithstanding, some solutions do not require the previous extraction of triggers to extract binary relations from text. Trigger recognition solutions can be developed based on rules, dictionary matching and machine learning.

2.4.4. Relation extraction

There are several techniques to perform extraction of relations from biomedical documents, such as based on co-occurrences, rules, linguistic processing, machine learning and knowledge. However, each approach presents its own advantages and limitations, revealing more appropriated for such tasks, considering the available resources.

- Co-occurrences: Co-occurrences assume that if two concepts are usually referred in a specific text passage, i.e., sentence, paragraph or section, they are related.
- Rules: Rule-based solutions apply pattern-based rules to extract relations between concepts.
- Linguistic processing: Linguistics-based approaches take advantage of the information provided by advanced linguistic parsers to automatically extract relations between concepts in scientific articles.
- Machine learning: Machine learning-based approaches represent a large share of the existent relation mining solutions. Such approaches apply relation extraction

as a classification problem. Thus, candidate relations are classified as being a relation or not.

- Knowledge: Knowledge-based approaches take part of knowledge bases to infer biomedical concept relations based on their profiles, which are built using relations from literature or from curated databases and/or ontologies.

2.5. Interactive Mining

Due to the high complexity of the biomedical domain and the ambiguity of the related scientific documents, automatic extraction of biomedical information remains a challenging tasks with several issues to overcome. Many of the state-of-the-art solutions already achieve high-performance results, however, gold standard results are still not achieved. The provided mistakes must be taken into consideration when updating existing knowledge bases or generating gold standard resources. Thus, one must carefully analyze the provided automatic information and correct the existing mistakes. As previously presented, various studies [8, 9] have shown that using automatic solutions to assist biocurators delivers improved curation times. Nevertheless, such solutions are still not being widely used by the scientific communities [10], which are the main target audience.

In order to properly build such solutions, it is important to understand the requirements of biocurators. In the previously referred survey conducted by Hirschman et al. [11], the authors also analyzed the requirements of annotators, regarding features and usability aspects essential to assistance resources, concluding that biocurators were more interested in resources that are easy to use, install and maintain by final users. More than a high-performance resource, biocurators required tools that give detailed feedback, such as ranked results and confidence scores, and provide easy to use features to export results in standard formats, as well as inline visualization of different levels of annotations. Thus, thinking on usability and user-friendly interfaces, it is important to develop interactive solutions that take advantage of automatic systems and existing knowledge resources to assist expert curators in their daily tasks. To do so, the interface with the curator is an important aspect that needs to be carefully analyzed for tool adoption.

BioCrative workshops [58, 59] have organized tasks in order to promote the development of interactive tools, which have been a fundamental milestone regarding the innovation on developing interactive solutions, the encouragement of collaborative work between biocurators and developers of TM tools. Many annotation solutions have been developed by several research groups, following different approaches, features and target tasks.

However there were two distinct tasks that were mainly approached by such solutions:

- Document triage: retrieve and rank relevant documents considering a specific goal, such as documents with high probability of containing PPIs;
- Information annotation: identify information contained in documents, such as concepts, relations and/or events.

Regarding the provided features, by analyzing state-of-the-art solutions in detail, we collected a brief list of general features provided by such tools:

- Comprehensive and self-explanatory visualization of documents and respective information;
- Document tagging and triage;
- Concept and/or relation annotation;
- Concept normalization;
- Automatic annotation services integration;
- Document comparison;
- Support standard formats in import and export features;
- Integrate existing resources for document retrieval;
- Active learning for triage and/or concept annotation;
- Search documents for terms and/or concepts.

Considering the target task and its requirements, researchers followed different solutions in terms of usability. The implemented solutions focused mainly on the following approaches:

- Desktop application: that may be developed for a particular Operation System (OS), may have some hardware requirements in order to work correctly, updates must be installed directly by the user and may also require some hardware upgrades to ensure that updates work;
- Web application: this solution does not need high-performance hardware to allow users to access the application, since they just have to use a web browser and work with resources available on the internet, including storage and Central Processing Unit (CPU) processing power;
- Web browser extension: this solution is based on an extension from the web browser, adding new functionalities through the web browser's Application Programming Interface (API), and requires different extensions depending on the browser. It also, requires that each user installs and configure it on their own web browser.

However, since the web-based applications suits better the user needs for being more easily accessible through a web browser and do not having any hardware or updates limitations, most of the approaches are web-based applications.

Brat

Brat [60] is one of the highly used online complete web-based solutions for interactive mining. This application supports inline document annotations, representations and integration. It also provides concept normalization, automatic services integration, search capabilities and document comparison. However, annotation task configuration (e.g., target concepts and relations, normalization resources, and automatic services) is considerably difficult and non-accessible for non-advanced users such as biocurators.

MyMinner

MyMiner [61] is a web-based solution for biocuration. This application supports concept tagging and normalization of a pre-defined set of concepts using a restrict set of previously processed resources. It also supports document triage, relation mining, integrates a service for automatic concept recognition, and document comparison.

Argo

Argo [62] offers workflow design options with previously built and integrated components. In this web-based application users are able to create custom processing pipelines for concept and relation annotation with manual correction, supporting multiple import and export formats. Even though such approach is powerful, creating such workflows may require advanced expertise and provides a level of flexibility that may not be comfortable for biocurators.

Other solutions, such as BioQRator[7], CellFinder[8], PubTator [63], RLIMSP[9], tagtog[10] and ODIN [64] follow typical web-based solutions with less usable user interactions and annotations representation, using tabular listings of concept and/or relation annotations with simple highlighting and sorting or scoring capabilities.

[7] http://www.bioqrator.org

[8] http://www.cellfinder.org/

[9] http://research.bioinformatics.udel.edu/rlimsp/

[10] https://www.tagtog.net

BioQRator

BioQRator is a web-based tool for annotating biomedical literature. This tool supports concept and relation annotation and it also supports the BioC[11] format [65] for input and output data. However, the interface provided for inserting and removing concepts or relationships stills does not follow the What You See Is What You Get paradigm, making the curation process harder to understand. On the other hand, BioQRator integrates solutions for document triage for protein-protein-interactions.

CellFinder

CellFinder is a web-based tool for biocuration, providing long-term data storage for validated and curated primary research data and provides additional expert-validation through relevant information extracted from text.

PubTator

PubTator features a PubMed-like interface with many state-of-the-art automatic solutions already integrated for concept recognition and normalization. Even thought this web-based tool uses advanced text-mining techniques, it still uses tabular listings of concept and/or relation annotations with simple highlighting and sorting/scoring capabilities, which makes annotated concepts more difficult to understand.

RLIMS-P

RLIMS-P is a rule-based text-mining program specifically designed to extract protein phosphorylation information on protein kinase, substrate and phosphorylation sites from biomedical literature. This tool works with PubMed abstracts and open access full text articles.

tagtog

tagtog integrates active-learning of concept names using annotated information, improving everyday performance with the previous results. This application supports several input and output formats, document editor and interactive mining. Additionally,

[11] http://bioc.sourceforge.net/

tagtog uses machine-learning techniques to improve the automatic predictions of annotations according to users' feedback.

ODIN

ODIN (The OntoGene Document INspector) is a tool aimed at supporting the curation of biomedical literature through integration of powerful text mining technologies. This web-based tool supports many input and output formats such as xml or plain text, and processes it with a custom NLP pipeline, which includes NER and relation extraction. However, the display of annotated concepts and relations still uses a tabular approach. This tool also provides concept normalization via UniProt [21], Entrez Gene [66], NCBI taxonomy [67], PSI-MI ontology, PharmGKB [68] databases.

SciKnowMine

SciKnowMine[12] (after 'Scientific Knowledge Mine') is a desktop application for document triage, which integrates active learning capabilities to obtain new models based on interactively annotated documents.

MarkerRIF

MarkerRIF[13] is a web-browser extension that allows annotating concepts directly on documents from the Pubmed web-site, providing relevant sentences retrieval and supporting normalization of a restrict set of concepts.

Overall, there are several interactive mining solutions that provide better experience to expert curators and help to improve their performance in some everyday tasks. However, existing solutions still have several issues or limitations that hinder the wider applicability and usability of these tools by expert curators. Those limitations are inherent to architecture, features, usability and performance:

- Architecture:
    - Lack of a flexible and ready to scale architecture to support new features and integration of new services.

[12] http://www.isi.edu/projects/sciknowmine/overview
[13] http://bws.iis.sinica.edu.tw/MarkerRIF

- Features:
    - Limited integration with existing major services for document retrieval;
    - Limited support to standard biomedical input and output formats;
    - Lack of integration with automatic annotation services and limited integration with existing state-of-the-art resources;
    - Limited visualization and/or interaction features;
    - Lack of an integrated and flexible configuration of target annotation information (e.g., users, annotation guidelines, concepts, relations and events);
    - No real-time collaboration features.
- Usability:
    - No WYSIWYG user interfaces with inline annotations and interactions, which difficult understanding, adding, tuning and removing annotated information;
    - Require advanced installation and configuration steps.
- Performance:
    - Slow document representation for visualization for large documents.

2.6. Evaluation

The accuracy of the automatic generated annotations can be measured in order to understand and evaluate the behavior of the developed solutions. This task is performed by annotating a corpus and then comparing the automatic results with the ones provided by expert curators. Therefore, each automatic annotation should be classified as:

- True Positive (TP): the annotation provided by the automatic systems exists in the manually annotated corpus;
- True Negative (TN): the nonexistence of a annotation is correct according to the manually annotated corpus;
- False Positive (FP): the annotation provided by the automatic system does not exist in the manually annotated corpus;
- False Negative (FN): the system does not provide an annotation that is present in the manually annotated corpus;

In order to obtain performance results and better understand the behavior of information extraction systems, exact and approximate matching can be used. With approximate matching we can find the performance when minor and non-informative mistakes are discarded.

Performance results are obtained using three important measures: precision, recall and F-measure. Those performance measures take values between 0 and 1. Precision measures the ability of a system to present only the relevant items, recall measures the ability of a system to present all relevant items and F-measure is the harmonic mean of precision and recall.

$$Precision = \frac{\text{relevant } items \text{ retrieved}}{\text{total items retrieved}} = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{\text{relevant items retrieved}}{\text{relevant items in collection}} = \frac{TP}{TP+FN} \tag{2}$$

$$F\text{ - }measure = 2\ \frac{Precision.Recall}{Precision+Recall} \tag{3}$$

On the other hand, there are other relevant measures for evaluating binary classification problems, such as accuracy, which measures the ability of the system to provide correct predictions (positive and negative), sensitivity which measures the ability of the system to provide positive results, and finally, specificity which measures the ability of the system to identify negative results [69].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{5}$$

$$Specificity = \frac{TN}{FP+TN} \tag{6}$$

2.7. Summary

This chapter presented a detailed analysis of the tasks of biomedical information mining required to perform extraction of concepts, relations and events from scientific literature. Moreover, another relevant aspects were specified, such as the importance of resources like knowledge bases and corpora in biomedical information extraction, advantages and limitations of each solution and the generic workflow of each technique. Otherwise, on the next pages we present Egas, a web-based collaborative platform for interactive biomedical literature curation that intends to minimize the aforementioned limitations, delivering a highly flexible, usable and easy to understand solution.

Chapter 3

Requirements

As we saw in the previous chapter, there are several features but also some limitations in the existing state-of-the-art solutions. Nevertheless, it is crucial to continue improving this field and develop new solutions that can help curators to do their work easily. Likewise, in order to create a useful and complete application that supports every need of biomedical curation, there are some requirements that we have to be aware of. This way, in the next few pages we will describe the user and system requirements that supported the development of this solution and that we aim to achieve.

3.1. User requirements

The underlying need for a tool accommodating several requirements was motivated by existing limitations of the available curation tools associated with the urge to improve the efficiency of biomedical documents' curation. Likewise, some of these limitations, as described in section 2.5, affect the performance of everyday tasks and hinder the wider applicability of these systems by expert curators.

Furthermore, curators require solutions that can agglomerate several features and improve their performance and efficiency in order to provide a better and consistent data curation. On the other hand, usability may be a key-factor to enhance the designed solution, since the existent systems lack in user-friendly interfaces and intuitive procedures. Thus, curators should be able to add and remove annotations with just a few clicks, and document processing and visualization must be fast and optimized. Nevertheless, interface design should also be a major requirement allowing curators to focus essentially on document and its annotations, simplifying as much as possible the system configuration and user settings.

Additionally, other requirements focus on performance and portability of the developed solution. On one hand, document rendering and visualization should be fast and optimized in order to load, in just few seconds, large documents with thousands of annotations. On the other hand, curators need a solution that, apart from providing several features, can also be accessed in almost every internet-capable device, saving unnecessary configuration times since it is delivered in a centralized way.

3.1.1. Mission

The goal of this solution is providing curators a better system regarding biomedical text mining and information curation. To do so, our system needs to deliver high-level performance and a user-friendly interface as well as a wider application of state-of-the-art services and features.

Automatic text mining systems provide efficient information extraction results, however, they are not sufficient to produce ready-to-consume curated documents. Even though these systems are not meant to replace curators, they can assist them in one or more biocuration tasks and help them improving their results. As a matter of fact, the interface and interaction with the curator appears to be an important aspect that needs to be considered and carefully designed in order to create a useful text mining tool [7].

In summary, this solution aims to provide curators a state-of-the-art text mining system for fast and accurate curation of information. By taking advantage of several implemented functionalities, aggregating automatic annotation services and delivering real-time collaboration features, we intent to facilitate the task of creating and updating knowledge bases and annotated resources.

3.2. Functional requirements

As we saw in the previous chapter, there are several solutions for biomedical text mining. However, some limitations were found in the available systems that need to be covered. Thus, the most important functional requirements that the desired solution must cover fit in four different areas:

- System requirements: overall system features and interface requirements;
- Visualization and text curation: which are associated with document representation and annotation;
- Project management: requirements associated with project administration and configuration;
- User management: where we list all requirements related with user accounts;

Regarding to system requirements, we need to essentially assure the management of users, projects and documents and their respective requirements:

- Support user account creation and respective account management;

- Create projects: user should be able to create and manage their own projects;
- Projects should be private, where only the associated users can see and perform changes in the documents, or public, where everyone has access to the project documents but only associated users can perform changes;
- Import and export documents and respective annotations in specific formats, namely BioC and A1 formats;
- Search (PMID[14] , PMCID[15] or keyword) and import documents from PubMed and PubMed Central;
- Support visualization of documents and their respective concepts and relations;
- Full-text processing: system should be able to display full-text articles;
- System must provide automatic annotation services to help curators improve their results;
- System must record curation time: be able to record time of curation for each user by article;
- Users should be able to search projects in the application;
- Users should be able to request invites to specific projects.

Visualization of documents and respective annotations is the main requirement of the desired system. Furthermore, curators should be able to easily add or remove concepts and relations from the document. On the other hand, the solution must provide automatic annotation services to help curators improve their results and retrieve import information from biomedical data in a more efficient way. Accordingly, there are several requirements related with document representation and curation.

- System should be able to display documents and respective annotations;
- Display inline concept annotations;
- Display inline concept relations;
- User should be able to add and remove concepts from the document;
- User should be able to add and remove relations between concepts;
- User should be able to turn on and off the visualization of the target concepts and relations;
- Normalization of annotated concepts using the provided ontologies;
- Users should be able to perform automatic annotation of concepts and relations on the desired documents with the provided services;
- Users should be able edit the automatic text mining results;

[14] **PMID** (PubMed identifier or PubMed unique identifier) is a unique number assigned to each PubMed record.

[15] **PMCID** (PubMed Central identifier) is a unique number assigned to works published in the free to access PubMed Central.

- Users must be able to perform automatic annotation of several documents at the same time;
- Users should be able to select the target concepts of the automatic annotated data;
- System must provide features to import and export various documents at the same time;
- Collaborative features: curators should be able to interact with other curators that are working on the same project. Every change made in a given document should be *broadcasted* to the other curators that are working on the same document;
- Users should be able to turn on and off the collaborative mode.

Project management and configuration combines several requirements that need to be covered. In order to offer a high-level solution to biomedical text mining, users should be able to configure their projects in their own way. Likewise, these are the requirements related with project management and configuration:

- Users should be able to manage their own projects;
- Projects should have different access levels:
    - Administrator: can annotate documents, manage the entire project and invite users;
    - Curator: only has document annotation permissions;
- Invite users to project: administrators should be able to invite other users to join the project;
- Define target concepts and target relations: administrators should be able to add, remove and edit the target concepts and relations. Likewise, should be associated to the target concepts a concept name and respective color. Nevertheless, administrators should also be able to select the desired ontologies to use for normalization purposes;
- Annotation guidelines: administrators should be able to add, edit and remove annotation guidelines which will be available for every user associated with the respective project;
- Project statistics: system must provide project statistics, namely curation times, number of concepts and number of relations by user or article;
- Remove articles: administrators should be able to remove articles from a project.

Additionally, users should be able to edit their account information. Likewise, these are the requirements regarding user management:

- Edit account name: users should be able to edit their accounts' name;
- Edit account password: users should be able to edit their accounts' password;
- Edit email address: users should be able to edit their email address;
- Manage associated projects: users should be able to see the list of project that they are associated with, the list of project requests and project invitations.

On the other hand, in order to provide a centralized solution, system administration features should also be implemented. Thereby, these are the requirements in terms of global system administration and management:

- Manage the list of projects;
- Block and remove projects;
- Add and index new ontologies;
- Remove existing ontologies.

3.3. Non-functional requirements

Despite all functional requirements listed in the previous section, there are other issues and limitations that our system needs to cover. Some of these non-functional requirements focus on usability, performance or even compatibility of the desired system. On the other hand, the system architecture is also a relevant factor in order to provide an efficient solution, ready to scale and prepared to support the development of new features.

In the next pages, the non-functional requirements of this solution are presented.

3.3.1. Architecture

Constantly evolving areas such as the biomedical domain comes up with new services and systems almost every day. Thus, the support of new services and respective implementation and integration with the desired solution must be one of the major requirements. Thereby, investing in a ready to scale, extensible and modular architecture, should be a priority in order to keep the system up to date and deliver to curators the state-of-the-art services in terms of automatic concept identification and annotation.

Nevertheless, other features, unrelated with automatic services, should also be easy to add and develop, thus, Egas architecture and data structure, should be designed thinking in flexibility and scalability, ready to support new features at any moment.

Additionally, the solution aims to deliver a complete and powerful service in terms of biomedical text curation. Thus, by delivering a centralized solution, we can avoid large configuration steps and provide a ready-to-use system, facilitating the setup and on-demand configuration of annotation tasks.

3.3.2. Performance

Other non-functional requirements rely on performance issues, such as loading of documents' information and their visualization in the browser window. These tasks should be performed the more quickly as possible in order to avoid long rendering loads and compromise the entire application usability. However, performing these tasks using standard web developing techniques, such as HTML[16], CSS[17] and JavaScript, can be more difficult than it looks like. Thus, the design and optimization of the rendering and document parsing algorithms is crucial to maintain the desired performance.

Furthermore, as far as we know, previous solutions take advantage of the capabilities of Scalable Vector Graphics (SVG) to display inline concepts and relations. Nevertheless, loading large documents, such as full-texts with thousands of concepts and relations, may take a considerable amount of time rendering the document and respective information using SVG. As a matter of fact, providing a solution that represents documents and respective annotation data, using standard web technologies, can achieve similar results and require much less time. Nonetheless, these techniques appear to be less flexible in terms of representation capabilities and, as a result, representing concept and relation annotations using such technologies requires the application of ingenious techniques.

3.3.3. Usability

Usability and user-friendly interfaces should be also an important requirement of a complete platform for scientific literature curation. As a matter of fact, a solution that can deliver an easy to use tool for biocuration can improve the curators results and, consequently, keep the existent knowledge bases updated in a more consistent way. On the other hand, usability issues may compromise the purpose of the entire solution, since providing a system that instead of helping curators will create some barriers to their work may be a compromising factor.

In summary, the application must be focused on usability, simplicity and user-friendly interaction, offering highly usable interfaces for manual and automatic in-line annotation of concepts and relations.

[16] **HTML** (HyperText Markup Language)

[17] **CSS** (Cascading Style Sheets)

3.3.4. Compatibility and portability

The system compatibility is another crucial non-functional requirement aiming to provide access to the developed solution in almost all internet-capable devices. Thus, the develop solution must be compatible with the major key-players on the market, namely Mozilla Firefox, Google Chrome, Internet Explorer and Safari.

On the other hand, system should support several different operating systems, since biocurators might have different preferences in this area. Thus, the application has to support being ported to other systems, interfaces, internet accesses.

3.4. Summary

In this chapter, the requirements that shaped this solution were presented. Regarding to system requirements, we conclude that major issues of existent solutions, such as architecture limitations and lack of integrated state-of-the-art services should be addressed and covered by the desired application. On the other hand, usability and fast document processing and visualization may also be a key-factor in providing a complete tool for biomedical literature curation. Thereby, the representation of inline annotations and relations, providing a "What You See Is What You Get" interface appears to be an important requirement and a major feature to deliver a high-level solution.

Finally, taking into consideration the target users and the overall system requirements, our solution must overcome the main limitations of existing systems, aiming at the development of a state-of-the-art text mining system for fast and accurate curation of biomedical literature.

Chapter 4

Architecture Proposal

Considering the previous listed requirements, we present here a proposal for an architecture that supports all the features that we need to implement. To create a fluid application that answers all the user requirements and that is, at the same time be, as user-friendly as possible, we need to choose very carefully the technologies that will support the core of the application, as well as the user interface, namely the visualization of the annotated concepts and relations. Nevertheless it is also crucial to find and choose the most advanced and optimized algorithms and techniques to avoid long page loads that might compromise and slow down the entire application.

This chapter presents the architecture of Egas, exploring the foundations of the application along with the problems and solutions that we had during its design.

4.1. Overview

Since our solution relies on a web-platform for biomedical data curation, our architecture is divided in two parts: client and server. The client-side is directly responsible for the interaction with the user, through their web-browsers, managing the visualization of projects, documents and the respective annotations. On the other hand, the server-side is responsible for all the processing and storage of the generated data. Likewise, the data is stored in a MySQL[18] database which is connected to a Java Web-Service. Finally, all the ontologies used for normalization purposes are stored and indexed using Apache Solr[19].

Figure 9 presents a general overview of the architecture. All data exchange between both sides of the application is made through a secure and encrypted channel using authenticated and authorized services.

[18] http://www.mysql.com/

[19] http://lucene.apache.org/solr/

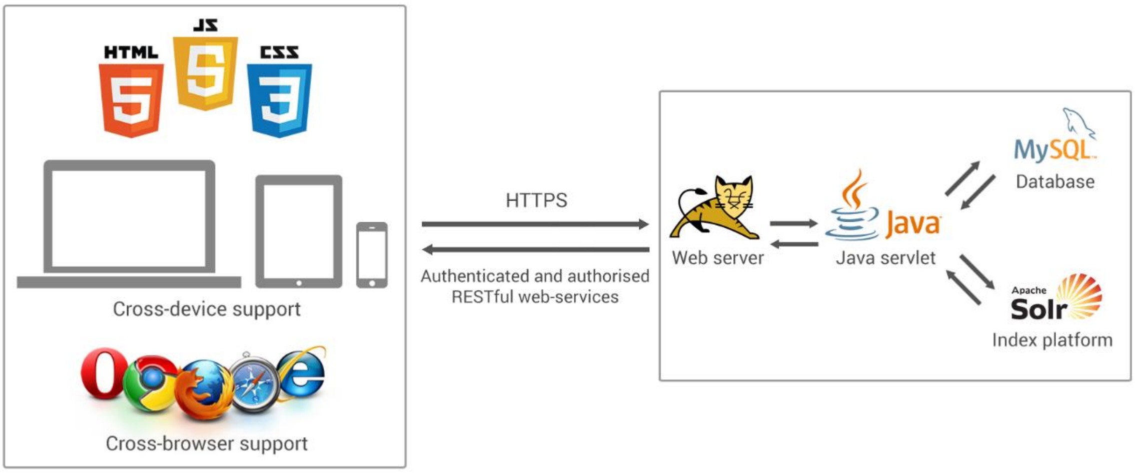


Figure 9: General overview of the implementation of Egas.

4.2. Client

When building a web-based application, one of the major concerns focuses on usability and user-interface. As we saw in the first chapter, the existing solutions in interactive mining presents some limitations regarding intuitive visualization of concepts and relations and require an advanced installation and many configuration steps. On the other hand, performance issues, such as slow document representation, should be taken in consideration, since they could compromise the usability of the entire application.

Our solution intends to fulfill these issues and limitations providing an innovative and flexible solution for biomedical data curation, since it is easily available for most devices with an internet connection. As a web-based platform and since document collections, users, annotations and back-end data storage are all managed centrally, the advanced installations and configuration steps are also avoided. On the other hand, performance and compatibility are key-factors that we must be aware of while developing our application. Thus, we choose standard web technologies, making Egas available in both desktop and mobile devices, supported by the most widely used web-browsers on the market.

This way, the web application uses HTML5 to describe and structure the document and CSS3 to describe the presentation semantics and formatting of that same content. Furthermore, in order to provide a greater user experience and interaction, we use JavaScript to manage the dynamic content, communicate asynchronously with the available services and update the document data and content.

These technologies allow us to create a widely supported application, however, it is very important to assure that the same standards work in the different browsers.

Nonetheless, even though these technologies deliver fast representation of information and cross-browser support, performance issues can emerge with large amounts of data. Thus, the development and implementation of fast and optimized algorithms are crucial to maintain a fluid application and enable fast load and visualization of documents. Moreover, a web application which includes all the user and functional requirements listed in the previous chapter can achieve a few thousands of lines of code on both server and client sides. Likewise, if we are building a single-page application[20] using JavaScript - like Egas – it is extremely important to use a robust and solid JavaScript framework to add structure to all the data, easily manipulate the document and avoid code repetition.

4.2.1. JavaScript

JavaScript plays an essential role regarding user interaction in web applications. By running on the client-side of the application, JavaScript tasks are processed and completed almost instantaneously, as they do not need to be processed in the server-side and sent back to the user, consuming local, as well as server, bandwidth and time.

Traditionally, web applications leave the heavy-lifting of data to servers that push HTML to the browser in complete page loads. This way, the use of client-side scripts was limited trying to improve the user experience. Nowadays, this relationship has been inverted, client applications pull raw data from the server and then render it into the browser when and where it is needed [70].

This way, creating a single-page application will reduce the number of server requests providing a fluid user experience and less waiting times since all the requests are made asynchronously. On the other hand, several JavaScript libraries helps developers creating powerful applications faster by adding some structure to the code. Thus, when designing our application, a wide range of libraries and frameworks were analyzed in other to choose the one that better fits our solution.

Backbone.js

Modern JavaScript frameworks can bring structure and organization to our projects, establishing a maintainable foundation right from the start [70].

One of the most powerful JavaScript frameworks is Backbone.js. Backbone allows us to structure the JavaScript code in a Model-View-Controller (MVC) architectural pattern which separates the concerns in an application into three parts:

[20] **Single-page application** (SPA) - also known as single-page interface (SPI), is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application

- Models represent the domain-specific knowledge and data in an application, like a User, a Book or a Movie. They can notify the other parts of the application when their state changes.
- Views are typically related with the user interface of the application (templates), but not in all frameworks. They keep track of changes occurred in Models in order to update the user interface.
- Controllers handle input data (e.g., clicks, user actions) and update Models.

This way, we can simplify the implementation of our application by creating different Views according to each feature/functionality and different Models to store the respective data. On the other hand, users input data through Controllers which update the data presented in Models. Views observe Models and update the user interface when changes occur. Nevertheless, some frameworks, like Backbone.js, merge the Controller responsibility into the View.

Since we are creating an application where much of the heavy lifting for view rendering and data manipulation will occur in the browser, Backbone.js is an indispensable framework which provides a minimal set of data-structuring (Models, Collections) and user interface (Views, URLs) primitives that are helpful when building dynamic applications. On the other hand, Backbone does not force you to stick to its structure, meaning you have the freedom and flexibility to build the best experience for your web application. You can either use the prescribed architecture it offers or extend it to meet your requirements. For these reasons, Backbone was the best choice for our JavaScript framework and our client-side foundations.

Client general architecture

Like every other application, we designed our solution to be modular and extensible in order to facilitate the addition of new features and functionalities as much as possible. Figure 10 presents the general architecture of our client-side interface, where we can see how every View interact with the other components of the application.

As we are creating a single-page solution, providing linkable, bookmarkable, sharable URLs for important locations in the application may sound difficult, however, with Backbone Routers we can provide methods for routing client-side pages with standard web URLs (e.g. /somepage), connecting them with actions and event handles in our JS code. Thus, we can use the Router component to navigate between projects and articles, keeping a browser history and a direct link for the document that we are currently seeing. As we see in Figure 10, the Router component is connected to every View in our application through the Event Bus.

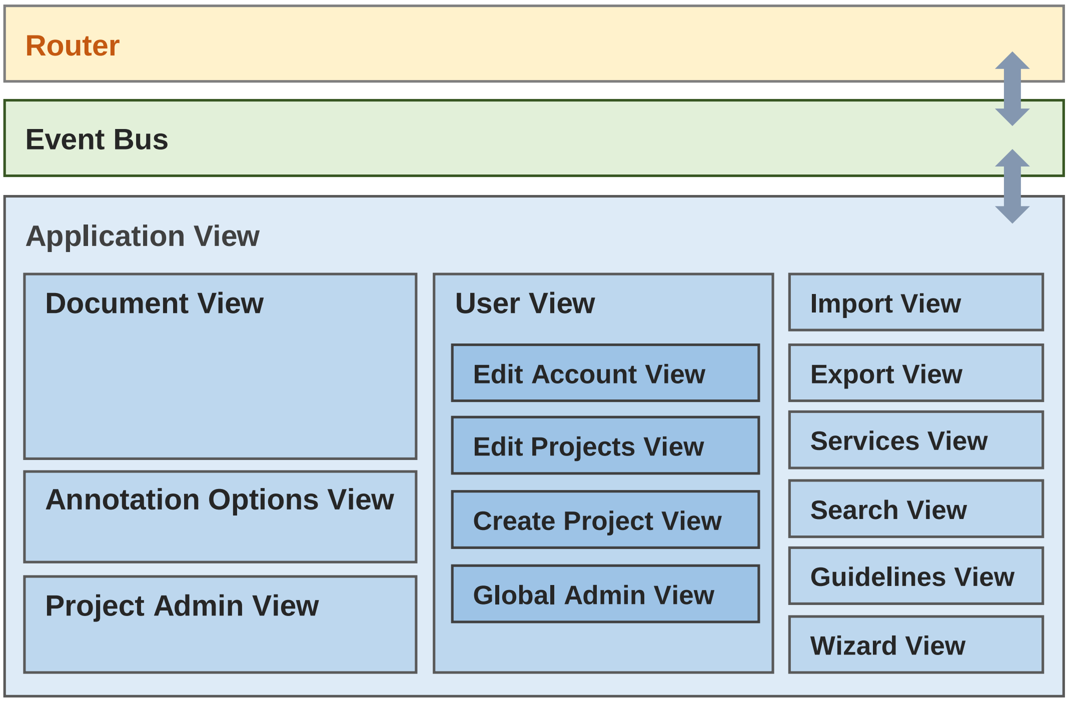


Figure 10: General client-side architecture of Egas

On the other hand, Backbone Events – represented as the Event Bus component in Figure 10 – helps us in the interaction with every other component in our project. Events allows us to trigger a function in another View without knowing exactly how it was implemented or even what it will do. We only need to know which event we need to trigger, and every View that was defined to handle this event will do their job. For example, when we change the project that we are working on, the Application View will trigger an event called, for instance, "project:load" which will be handled by Annotation Options View in order to update the target concepts and relations according to the new selected project. In a web application like Egas – with constant change of data – Events are very useful, since we can trigger multiple functions at the same time, updating different parts of our application to keep all the data and information synchronized.

As we can see in the bottom of Figure 10, our architecture is composed by several views responsible for the manipulation and rendering of all the stored data. The main container – Application View – handles all the system configuration and interaction with the user regarding project configuration and administration. On the other hand, all interaction between the document and the curator, like rendering the article annotations, annotating concepts or even adding and removing relations are handled by the Document View, which is directly connected with the Annotation Options View in order to keep all concept data updated. Project Admin View, allows project administrators to configure their projects, invite users, manage the target concepts and relations and access project statistics. Likewise, User View is responsible for managing user account settings like edit account information, manage user projects and, if the user has global administration permissions, the management of the entire application.

Finally, there are some other Views responsible for other functionalities related with project management and administration, like import and export documents, project guidelines and project wizard, annotation services, where we can perform automatic identification of specific concepts and/or relations in a custom set of documents, and search for existing projects in the application.

4.2.2. User interface

Our application is designed to be simple and easy to use, providing a comfortable user experience while helping curators annotate biomedical texts faster. In order to create a fluid interface, highly focused on the representation of document and respective annotations, we take advantage of several high-end technologies regarding to front-end developing and templating, such as:

- **Bootstrap:** Bootstrap is the most widely used framework for front-end developing, mixing HTML and CSS-based design templates for typography, forms, buttons and many other interface components. It also contains several JavaScript extensions providing easy-to-use dialogs boxes, tooltips, modals and other interaction windows. Additionally, the implemented components and features are cross-platform and widely supported by the mostly used web browsers in the market.
- **jQuery:** jQuery[21] is a JavaScript framework designed to simplify the client-side scripting of HTML. As of 2014, jQuery is behind over 80% of the most visited websites[22], making it the most popular JavaScript library in use today[23]. In our application, we use jQuery to easily navigate in the document, select DOM elements, create animations, as well as connect asynchronously with the web server, through AJAX[24] calls.
- **Handlebars:** Handlebars is a templating JavaScript library that helps developers to easily populate their user interfaces. Working as a superset of Mustache template system, Handlebars adds extensibility and minimal logic which helps developers create more intuitive templates.

[21] http://www.jquery.com/

[22] http://www.similartech.com/categories/javascript

[23] http://w3techs.com/technologies/overview/javascript_library/all

[24] **AJAX** (Asynchronous JavaScript and XML) group of web development techniques used on the client-side to create asynchronous web applications, making them capable of asynchronously (in the background) sending/retrieving data to/from a server.

Mixing these technologies along with other ingenious techniques to improve and boost the application performance we are able to create a high-level platform for biomedical data curation and text mining which will help curators easily and more effectively keep current knowledge bases updated.

4.3. Server

The server-side of the application is responsible for storing all the data in a unique resource. Moreover, it also provides services for the application interaction with that same data. Therefore, we need to design an architecture capable of storing all information and, at the same time, providing quick access to data and be of easily integration with every standard application, such as web, desktop or mobile.

Therefore, to store all the application data, namely users, projects, the respective documents, annotations and configurations we use a MySQL relational database, where we can easily retrieve all the information that we need and keep the data consistent over time. Nevertheless, we need to create the methods and services that will handle and manage all the data to keep a secure and homogeneous access to all the application information. Thereby, we created a RESTful[25] web-service, developed in Java and deployed and made publicly available using an Apache Tomcat web server.

REST web-services provide us easy and fast access to the stored information along with simple integration with any development platform. RESTful APIs[26] are typically defined with a base URI[27] (e.g., http://application.com/resources/), an Internet media type (e.g., JSON[28], XML[29]) for the returned data and with standard HTTP methods (e.g., GET, PUT, POST or DELETE). Table 4.1 presents how the HTTP methods are typically used in a RESTful API.

Thus, using a RESTful web-services we can provide several methods allowing an appropriated access to the application data, at the same that we grant a secure and controlled way for the client-side application to exchange data with the server. On the other hand, the application data and information are very sensitive. Thus, we need to carefully control the access to the web-service methods and hence, to the database. Therefore, and since we have users with different levels of permissions, it is crucial to provide a role based access control, filtering the access to specific methods according to the user permissions. This way, we can secure our data, preventing possible attacks and assuring that every method and every change in our database is performed by the correct

[25] REST – Representational state transfer
[26] API – Application programming interface
[27] URI – Uniform resource identifier
[28] JavaScript Object Notation (http://json.org/)
[29] Extensible Markup Language (http://www.w3.org/XML/)

user and using the right means. The security and role based access was developed using Spring[30] Security that focuses on providing both authentication and authorization to Java applications. Further analyzes and explication of this security system will be described below in the 5.3.4 section.

Resource URI	http://application.com/resources/projects/**id**
GET	Retrieve the representation of the project with the requested **id** expressed in an appropriate Internet media type.
PUT	Update the data of the project with the respective **id**.
POST	Create a new project with the sent data (generally used without the **id** in the URI).
DELETE	Delete the project with the respective **id.**

Table 4.1: How HTTP methods are typically used to implement a RESTful API.

Additionally, and in order to guarantee complete protection of exchanged data, the communication between client and server sides is performed using a secured and encrypted channel using Hypertext Transfer Protocol Secure (HTTPS).

4.3.1. Services

Providing authenticated and secure access to the stored information is a key factor to deliver a trustful application. Thus, the development of a secure web-service that delivers the necessary methods to manage the stored information is a very thorough and important job. Therefore, we carefully designed and developed a Java web-service using Jersey[31], which is an implementation of JAX-RS (Java API for RESTful Web Services) that provides support for creating web-services and simplifies the development and deployment of web service clients and endpoints. Moreover, the developed methods allow us to manage all the data stored on the database, as well as manage all the indexed ontologies and respective associations with stored concepts for normalization purposes.

The developed services are deployed and made publicly available using an Apache Tomcat web server and are divided in four distinct resources:

1) Projects: this resource provides methods for managing all projects information and features, namely project configuration, target concepts and relations or even import and export documents;

[30] http://projects.spring.io/spring-security/
[31] https://jersey.java.net/

2) Documents: this resource provides methods for managing all information regarding to documents, such as annotated concepts and relations;
3) Users: this resource provides methods for managing user accounts, namely create and invalidate sessions, create new accounts, edit account information and invite, request and associate users to projects;
4) Admin: this resource, available only for system administrators, provides methods for managing Egas's platform, such as index new ontologies in Apache Solr platform, manage the available ontologies and manage all application characteristics.

4.3.2. Data structure

With all information centralized in a unique resource, we can provide a ready-to-use annotation platform for biomedical curation. Thus, in order to store all information related with projects, documents and users, data structure was created – presented in Figure 11 – designed thinking on flexibility and scalability of the application and its data.

Our structure allows each project to have multiple users, specifically project administrator and curators, which are responsible for the general interaction with the system. Likewise, each project is associated with its respective documents and description of annotation guidelines, which can be provided as attachment files. Moreover, every project may contain multiple target concepts and relations for annotation, represented in Figure 12 as meta concepts and relations. Each target concept and relation is defined by a specific name and representation color. Meta relations also have a direction type associated, which can be unidirectional, bidirectional or without any specific direction, in order to cover all possible cases.

Thereby, users can annotate concepts and relations in specific documents. Every concept is associated with a start and end character positions as well as with its contents. On the other hand, every annotated relation considers two target concepts and an associated directional type (Figure 13). Additionally, to every target concept we can associate a target ontology to perform concept normalization. Thus, Figure 12 and Figure 13Figure 14 presents a data structure designed to offer normalization features allowing curators to associate a unique identifier to every annotated concept. Furthermore, since all normalization data is indexed and stored using Apache Solr, the Normalization table present in Figure 12, contains information about every ontology allowing us to associate the respective Solr core to the desired ontology.

Finally, in order to keep track of the time spent by each user while annotating biomedical documents, Egas records the elapsed curation time of every user by document.

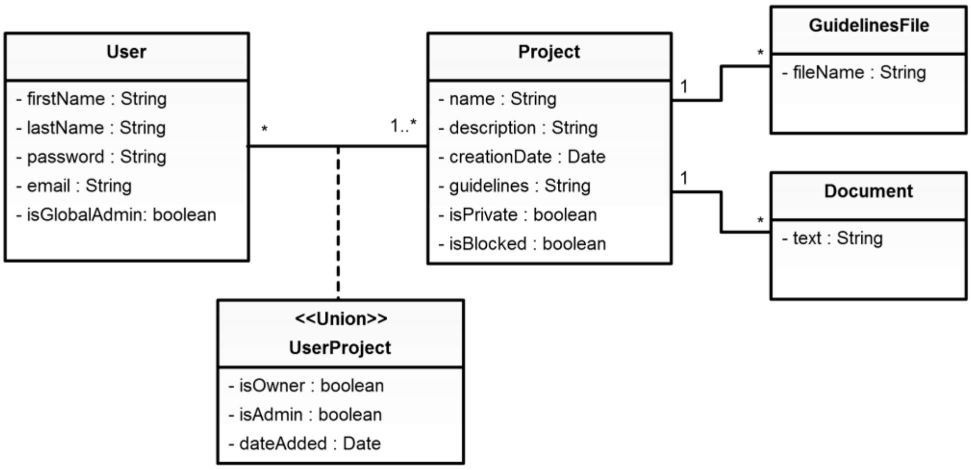


Figure 11: Project data structure.

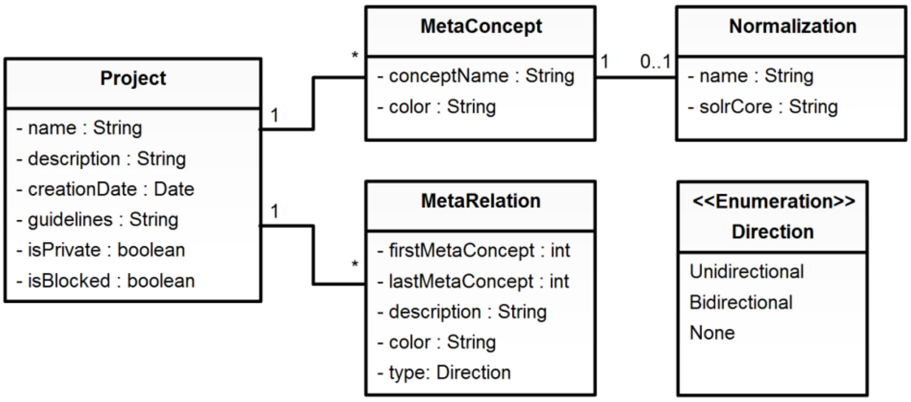


Figure 12: Meta-annotations data structure.

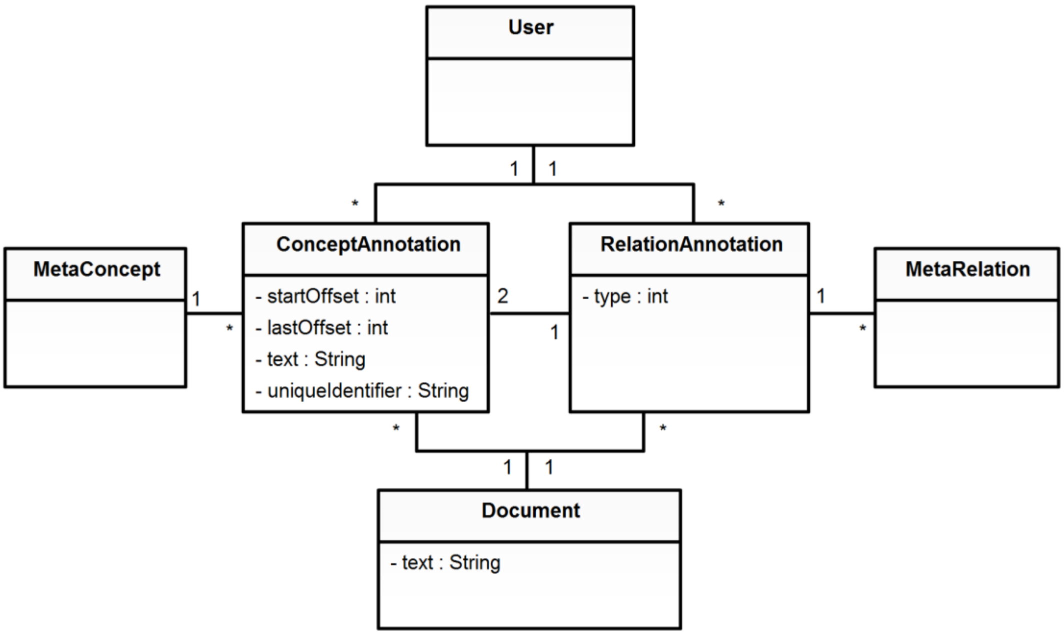


Figure 13: Document annotations data structure.

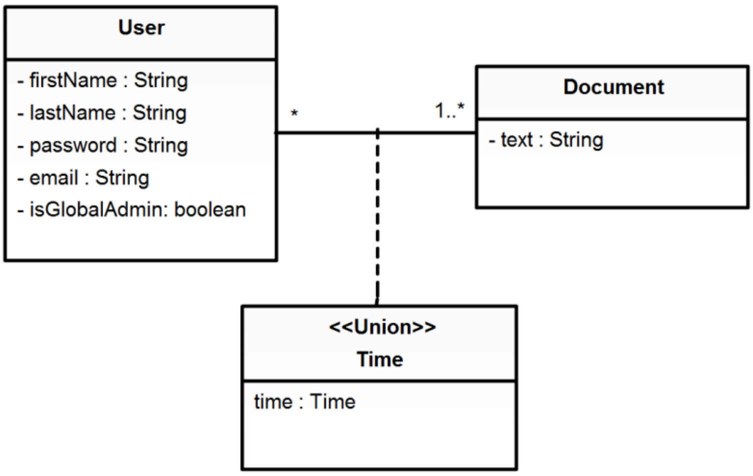


Figure 14: Curation time data structure.

4.3.3. Normalization

In order to offer normalization features in the easiest and fastest way as possible for biocurators, we indexed and integrated a rich set of biomedical knowledge bases. As one of the most advanced tools for indexing, storing and searching through different kinds of text collections, Apache Solr was used to index the identifier, preferred name, synonyms and definition (if available) of each concept in these resources. Furthermore, targeting flexibility and robustness, a separate index is used for each knowledge base.

Additionally, since knowledge bases are available in heterogeneous formats, we developed scripts to automatically index ontologies in OBO and OWL formats. To provide a better and centralized user-experience, these resources are integrated and available in the user interface where the system administrators can easily manage the existing ontologies. On the other hand, resources available in custom formats require the development of custom parsing algorithms.

In order to cover the wide spectrum of biomedical knowledge, we decided to collect ontologies provided by OBO Foundry [71]. Thus, a total of 110 ontologies were indexed, including NCI thesaurus [72], NCBI taxonomy [73], Protein Ontology [74], Gene Ontology [22], ChEBI [75] and Disease Ontology [76]. Overall, more than 2 million entries are indexed and available for biocurators.

4.4. Real-time collaboration

Real-time collaboration features are implemented by taking advantage of TogetherJS[32] from Mozilla, which is a JavaScript library built on top of Node.js[33] to simplify the development and interaction with collaboration and multiplayer features. Node.js enables broadcasting users' actions to all active users (Figure 15), maintaining a listening room in a server, and when a user performs an action, the details of that action are sent to the room, so the room can broadcast the action to all other users, which process the action accordingly. That way, each document has one collaboration room, so all active users can observe the actions performed by other users. Moreover, we defined that the actions of adding, changing and removing both concept and relations are sent to the collaboration room and broadcasted to other users. Additionally, every project have a dedicated chat, allowing users that are annotating different documents to discuss details of the annotation guidelines in order to minimize as much as possible the performed mistakes.

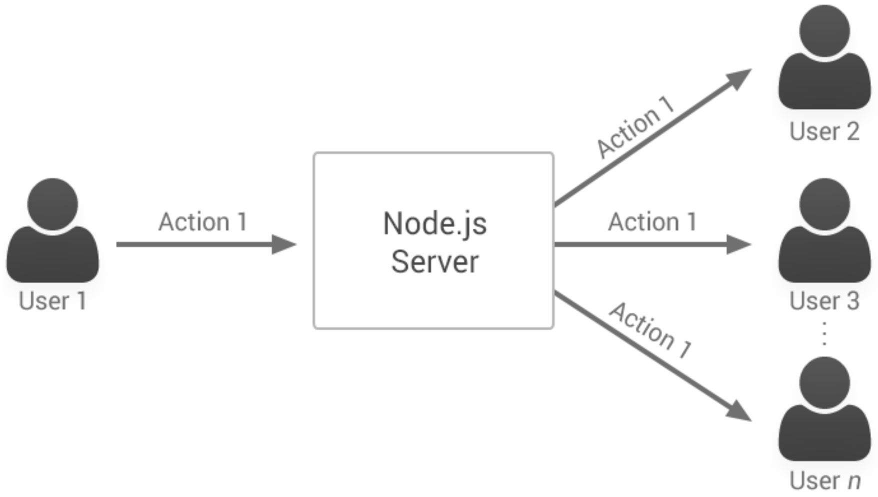


Figure 15: Illustration of the technique applied by Node.js in order to enable real-time collaborative features.

4.5. Summary

This chapter presented the foundations that sculpted our solution. By taking advantage of a centralized system, we can provide a ready-to-use service for interactive biomedical information curation. Furthermore, with the applied solutions and techniques our system is easily available for almost all internet-capable devices, delivering it a highly flexible, usable and easy to understand application. Overall, with the proposed architecture, we can provide a real-time collaborative system and enhance

[32] https://togetherjs.com/
[33] http://nodejs.org/

the communication between curators in order to achieve more consistent results. On the other hand, based on the provided features, such as manual and automatic annotation, guidelines and easy-to-use project configurations, we strongly believe that Egas is a state-of-the-art solution to perform different biocuration tasks, helping curators keep the current knowledge bases properly update and consistent.

Chapter 5

Implementation

In this chapter we describe the implementation of Egas, describing each feature and the general usage pipeline of the application. First, we present a system description and user interface and afterwards we describe the available features and the detailed information regarding its implementation and applied technologies.

5.1. System description

Egas is a web-based platform for text mining and curation which supports collaborative features. This web application allows users to create projects, import documents and annotate concept occurrences and relations between them. Since it is a web-based platform, Egas stores all the data centrally allowing a curation team to use the service, configured according to their preferences and taking advantage of a collaborative application. The developed solution was designed with a strong focus in usability and simplicity, in order to improve the efficiency and speed of biomedical text curation.

In the developed application, users can create projects (Figure 16). A project consists of a curation or document annotation task, performed on a collection of documents, by a team of curators, and considering a pre-defined set of concept and relation types defined by the curation guidelines. The project administrators are responsible for managing the curators associated to the project and the project characteristics, such as annotation guidelines and target concepts and relations. Furthermore, the project administrators are also able to associate the available ontologies with the target concepts in order to perform concept normalization.

Projects can be public or private, which are only accessible by users that have been added by the project manager. Therefore, a user can only annotate a document if he/she is properly logged in and associated with the respective project. Additionally, the system records all user operations, regarding to adding, editing or removing annotations and relations and also registers the curation time of each user per document. These statistics are available to the project manager in order to keep track of the annotation process per article and user.

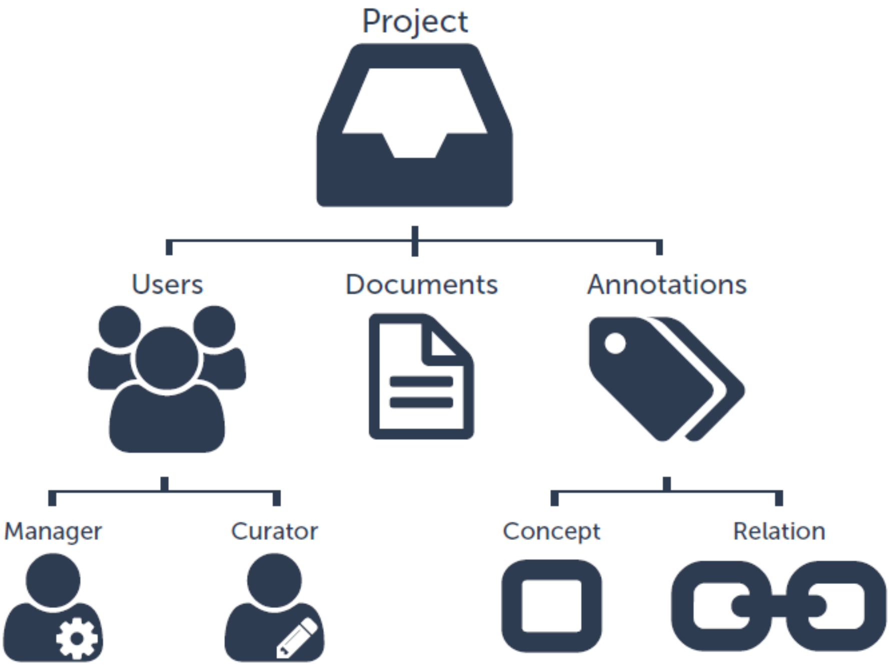


Figure 16: Organization of Egas based on projects, users, documents and respective annotations.

Figure 17 presents the typical usage pipeline of Egas and an overview of the provided features. In order to create a collection of documents, users may import documents from their devices or from known remote resources. Standard formats, such as A1 and BioC are supported, as well as plain text files. Furthermore, users are able to use remote resources to import documents with specific identifiers previously collected (PMID and PMCID), or by executing keyword search queries on PubMed or PubMed Central databases to select documents from the provided results.

Once they have the documents to annotate, curators can start from raw text and add concepts or relations annotations as they review the documents. On the other hand, they can also start by importing preprocessed texts, containing automatically identified annotations that they should revise. Furthermore, users can use the concept and/or relation extraction services, provided by the developed system, to pre-process a set of raw documents in the collection. These services are integrated with Egas in a flexible way, allowing the inclusion of new services for different purposes.

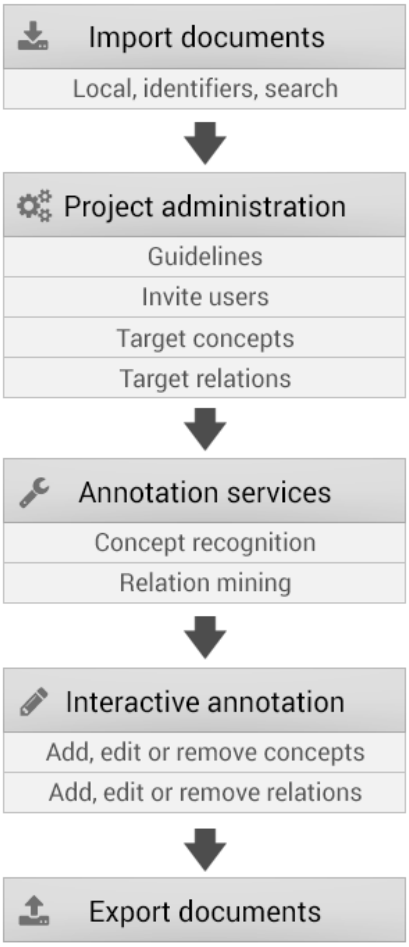


Figure 17: Typical usage pipeline of Egas presenting the provided features.

Moreover, project administrators can freely define the target concepts and relations types to annotate, according to the annotation guidelines of the tasks. In order to improve the results and facilitate the annotation work, each different concept and relation type is associated with a markup color. Relation types can be defined by specifying two concept types and assigning a description to the relation. For instance, for protein-protein interactions, the user should define a relation description (for example, “Interacts”), and choose the concepts types, in this case, both “Protein”.

Egas also provides real-time collaboration features, allowing instant feedback of users’ interactions within a document, such as adding, removing and/or changing concept and relation annotations. Thereby, multiple curators can change one document at the same time, showing exactly who change what to all users. Furthermore, a project chat is also available, providing a discussion room where users can share the details of the annotation task.

Finally, users are able to export all documents and the respective annotated data (concepts and relations) to standard formats, such as A1 and BioC.

5.2. User interface

The user interface is one of the most relevant factors of the application. The usability of the system, as well as the user interactions should be as intuitive as possible in order to help curators improve their work. In this section we present the relevant details of Egas's user interface and respective interactions. Furthermore, the main features and function characteristics are described in detail, namely the workspace, concepts and relation annotation, import and export documents, automatic services, project management and real-time collaborative features.

5.2.1. Workspace

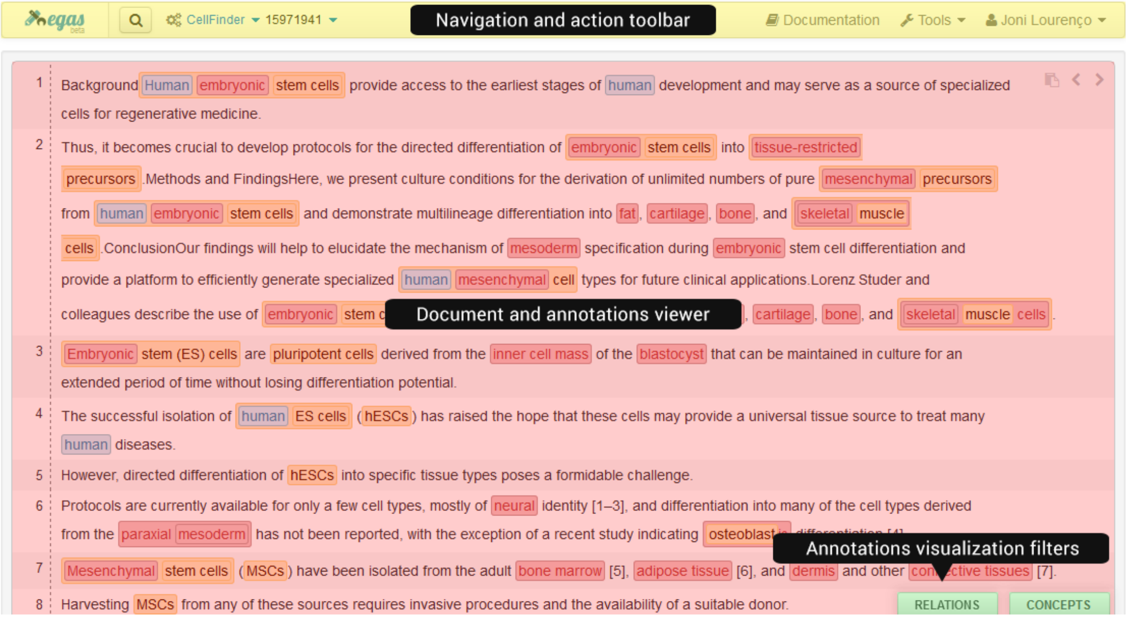
Egas was designed to be simple and easy to use, providing a user-friendly interaction, highly focused on the representation of the document and respective information. The main interface of the application contains only three main interaction areas (Figure 18):

- Navigation and action toolbar: provides the ability to navigate through projects and respective documents, search for existing projects and access integrated processing tools;
- Document and annotations viewer: presents the document with in-line concept and relation annotations and contains the main interaction area where the curator can add, edit and/or remove concept and relation annotations;
- Annotations visualization filters: enables filtering concepts and relations presented in the document viewer.

Moreover, in each interaction area, we can find the components that provides access to Egas's features, such as (Figure 19):

- Project management: manage and access project configurations, namely users, concepts, relations and annotation guidelines and statistics;
- Project and document navigators: navigate through users' projects and respective documents;
- Processing tools: access integrated processing tools, such as import, export and automatic annotation services;
- Account management: manage account information and the projects associated with the user;
- Document switcher: easily switch between documents in a project;
- Concepts visualization filter: select concept annotations presented in the document viewer;

- Relations visualization filter: select relation annotations presented in the document viewer.


egas beta
CellFinder
15971941
Navigation and action toolbar
Documentation
Tools
Joni Lourenço
Document and annotations viewer
Annotations visualization filters
RELATIONS
CONCEPTS

1 Background Human embryonic stem cells provide access to the earliest stages of human development and may serve as a source of specialized cells for regenerative medicine.

2 Thus, it becomes crucial to develop protocols for the directed differentiation of embryonic stem cells into tissue-restricted precursors .Methods and FindingsHere, we present culture conditions for the derivation of unlimited numbers of pure mesenchymal precursors from human embryonic stem cells and demonstrate multilineage differentiation into fat, cartilage, bone, and skeletal muscle cells .ConclusionOur findings will help to elucidate the mechanism of mesoderm specification during embryonic stem cell differentiation and provide a platform to efficiently generate specialized human mesenchymal cell types for future clinical applications.Lorenz Studer and colleagues describe the use of embryonic stem c cartilage, bone, and skeletal muscle cells.

3 Embryonic stem (ES) cells are pluripotent cells derived from the inner cell mass of the blastocyst that can be maintained in culture for an extended period of time without losing differentiation potential.

4 The successful isolation of human ES cells (hESCs) has raised the hope that these cells may provide a universal tissue source to treat many human diseases.

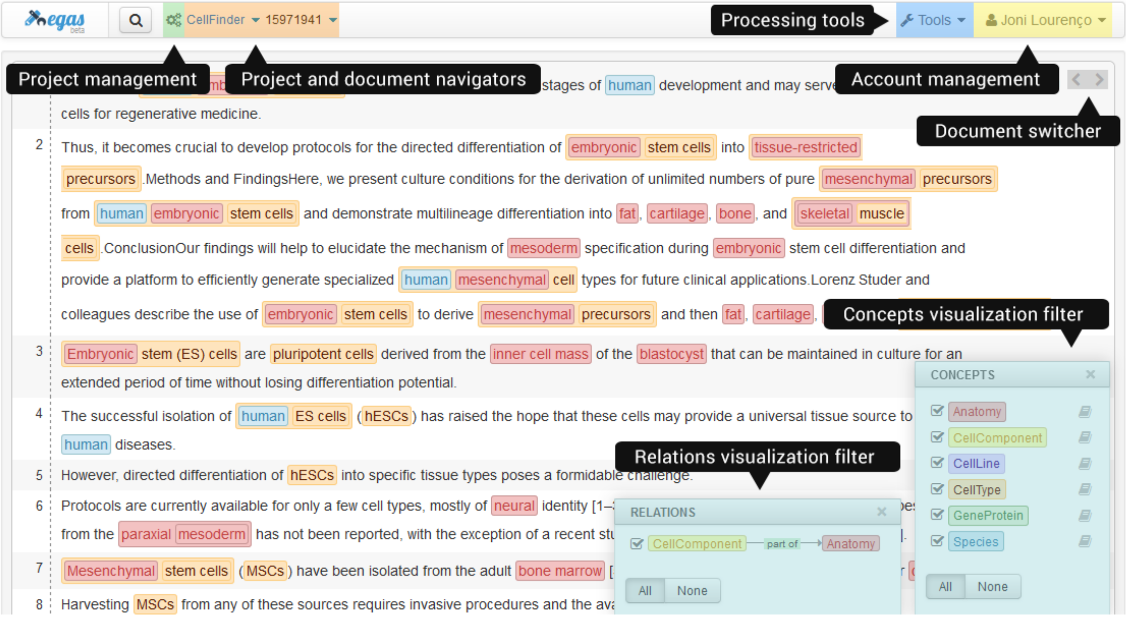
5 However, directed differentiation of hESCs into specific tissue types poses a formidable challenge.

6 Protocols are currently available for only a few cell types, mostly of neural identity [1–3], and differentiation into many of the cell types derived from the paraxial mesoderm has not been reported, with the exception of a recent study indicating osteoblast

7 Mesenchymal stem cells (MSCs) have been isolated from the adult bone marrow [5], adipose tissue [6], and dermis and other con ctive tissues [7].

8 Harvesting MSCs from any of these sources requires invasive procedures and the availability of a suitable donor.

Figure 18: Egas's main user interface interaction areas.



stages of human development and may serv
cells for regenerative medicine.

2 Thus, it becomes crucial to develop protocols for the directed differentiation of embryonic stem cells into tissue-restricted precursors .Methods and FindingsHere, we present culture conditions for the derivation of unlimited numbers of pure mesenchymal precursors from human embryonic stem cells and demonstrate multilineage differentiation into fat, cartilage, bone, and skeletal muscle cells .ConclusionOur findings will help to elucidate the mechanism of mesoderm specification during embryonic stem cell differentiation and provide a platform to efficiently generate specialized human mesenchymal cell types for future clinical applications.Lorenz Studer and colleagues describe the use of embryonic stem cells to derive mesenchymal precursors and then fat, cartilage,

3 Embryonic stem (ES) cells are pluripotent cells derived from the inner cell mass of the blastocyst that can be maintained in culture for an extended period of time without losing differentiation potential.

4 The successful isolation of human ES cells (hESCs) has raised the hope that these cells may provide a universal tissue source to human diseases.

5 However, directed differentiation of hESCs into specific tissue types poses a formidable challenge.

6 Protocols are currently available for only a few cell types, mostly of neural identity [1– from the paraxial mesoderm has not been reported, with the exception of a recent stu

7 Mesenchymal stem cells (MSCs) have been isolated from the adult bone marrow [

8 Harvesting MSCs from any of these sources requires invasive procedures and the ava

RELATIONS

☑ CellComponent part of Anatomy

All None

CONCEPTS

☑ Anatomy
☑ CellComponent
☑ CellLine
☑ CellType
☑ GeneProtein
☑ Species

All None

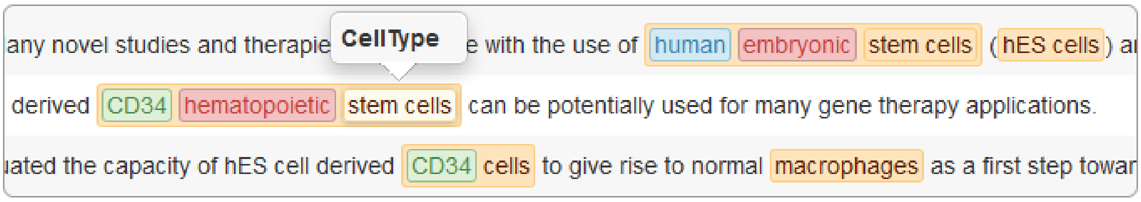
Figure 19: Egas's main user interface action components.

The display of annotation information, such as concepts and relations is performed inline the document. This way, curators can improve their annotation process since the application provides WYSIWYG[34] interactions giving instant feedback of the information

[34] What You See Is What You Get

added to the document. Concept annotations are highlighted in the document, with coloured boxes specific for each concept type, as defined by the project manager. Furthermore, due to the complexity of the biomedical domain, nested concept names may occur. Thus, they are carefully represented through superimposed coloured boxes (Figure 20) keeping an intuitive perception of the annotated information.

Visualization filters can also be applied in order to simplify document analysis. Thereby, users can select which concept type they want to be displayed and the ones that they want to temporally hide from the document. By unchecking the checkbox associated with a specific concept, Egas removes the coloured boxes around the annotation from the document viewer, simplifying the document representation and providing a focused analysis.



any novel studies and therapie e with the use of human embryonic stem cells (hES cells) a

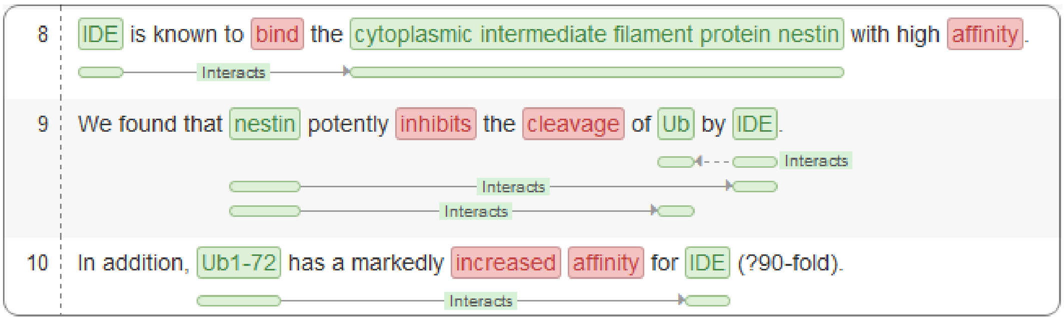
derived CD34 hematopoietic stem cells can be potentially used for many gene therapy applications.

ıated the capacity of hES cell derived CD34 cells to give rise to normal macrophages as a first step towar

Figure 20: Inline concept visualization and nested concepts.

On the other hand, relations are displayed below each sentence using directional lines between the two concept types (Figure 21), which is an innovative and easy to understand approach. Thus, coloured box are placed below each concept of the respective relation, with the same colour as the corresponding concept, which is connected by the relation line. Finally, the relation description goes in the middle of the relation line providing an easy perception of the presented relation. However, if the description does not fit in the space between the two concepts, it is placed on the right or left side of the relation depending on its position on the screen.

Additionally, when the users' cursor is over the relation, the application highlights the respective concepts in order to simplify the relation perception. Selective visualization of relations is also supported, applying a strategy similar to the concepts visualization filtering.



8 IDE is known to bind the cytoplasmic intermediate filament protein nestin with high affinity.

9 We found that nestin potently inhibits the cleavage of Ub by IDE.

10 In addition, Ub1-72 has a markedly increased affinity for IDE (?90-fold).

Figure 21: Inline relation annotations visualization with bi-directional relations.

Project and document navigation is also an important feature. Thus, we provide a fluid interface allowing users to easily switch through their projects and respective documents. Project navigation, available in the project navigator component (Figure 19), lists the projects associated with the user. The list contains the public (🔓) and private (🔒) projects that are accessible to the user. Likewise, feedback of projects where user is administrator (⚙) is also provided (Figure 22).

On the other hand, document navigation is available using two different approaches: the document navigator and the document switcher. Thereby, the document navigator provides the list of documents available in the project, allowing the user to choose a specific document of the project (Figure 22). Moreover, and since the number of documents associated with a specific project may achieve a considerable number, the document navigator loads more documents as the user scrolls down the list, preventing long and unnecessary rendering times. Furthermore, the document switcher (Figure 22) allows easy and fast document switching, by rapidly changing to the previous (❮) and next (❯) document, if available. This feature is also available with the directional keys on the users' keyboard.

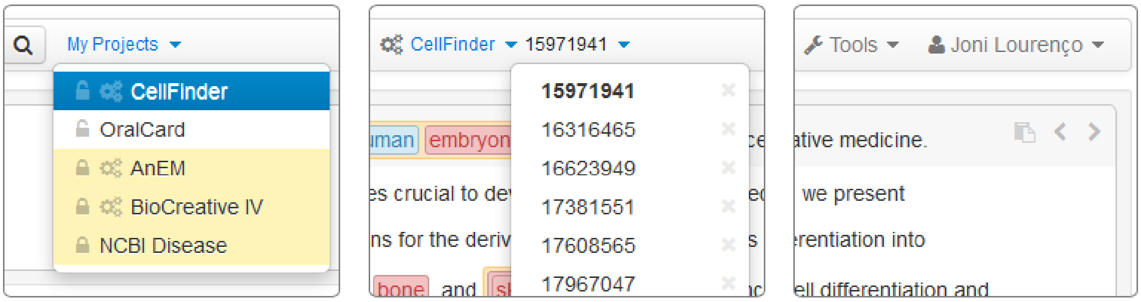


Figure 22: Project navigator (1), document navigator (2) and document switcher (3).

Additionally, users are able to search for public projects existing on the platform and find interesting topics to annotate. As previous described, public projects and respective documents can be accessed by anyone in Egas's platform, however, only users associated with the corresponding project can manage the respective annotations. Nevertheless, users can request an invite to a specific project in order to be part of the annotation process. Thus, once the project administrator approves the request, the project becomes available in the users' workspace and he can start annotating the respective documents.

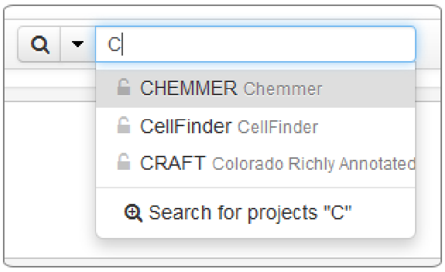
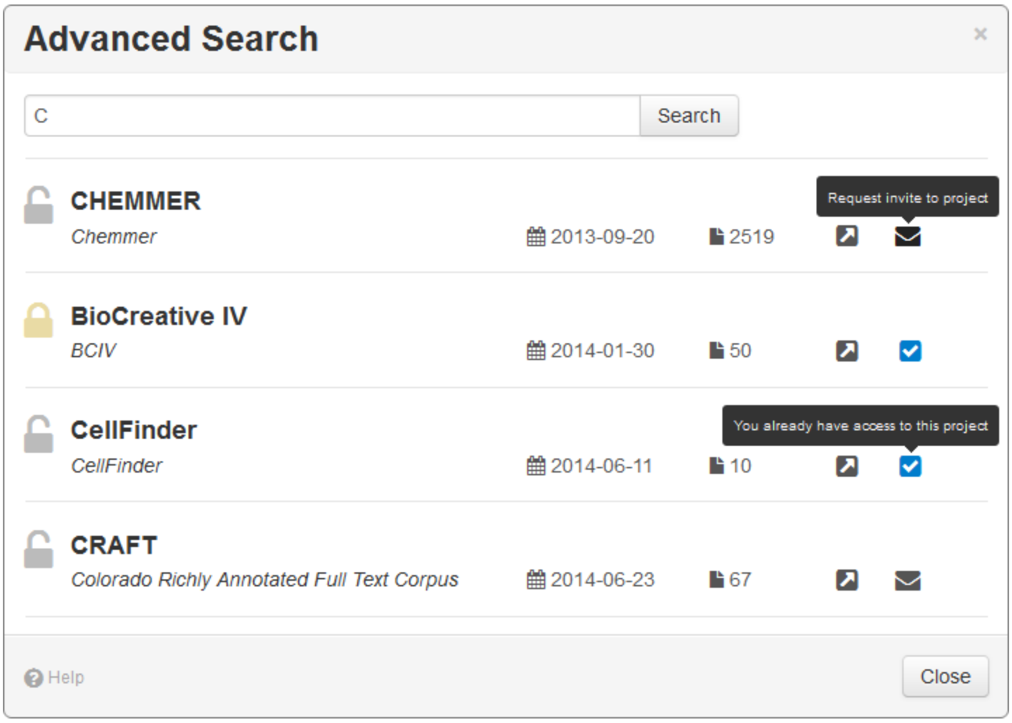


Figure 23: Project search box with suggested results.

The project search feature is available by clicking on the magnifying glass button to the left of the project navigator (Figure 23). Therefore, users can search by keyword using the *typeahead* input field provided. Nevertheless, users can find more results by clicking on the advanced search button (last item in the suggestion list). On the other hand, in the advanced search window, users are able to see more information about the project search results and request permission to the desired projects (Figure 24). Additionally, when requesting an invite to a certain project, users need to write a message informing the project managers of their intension.



Advanced Search

C

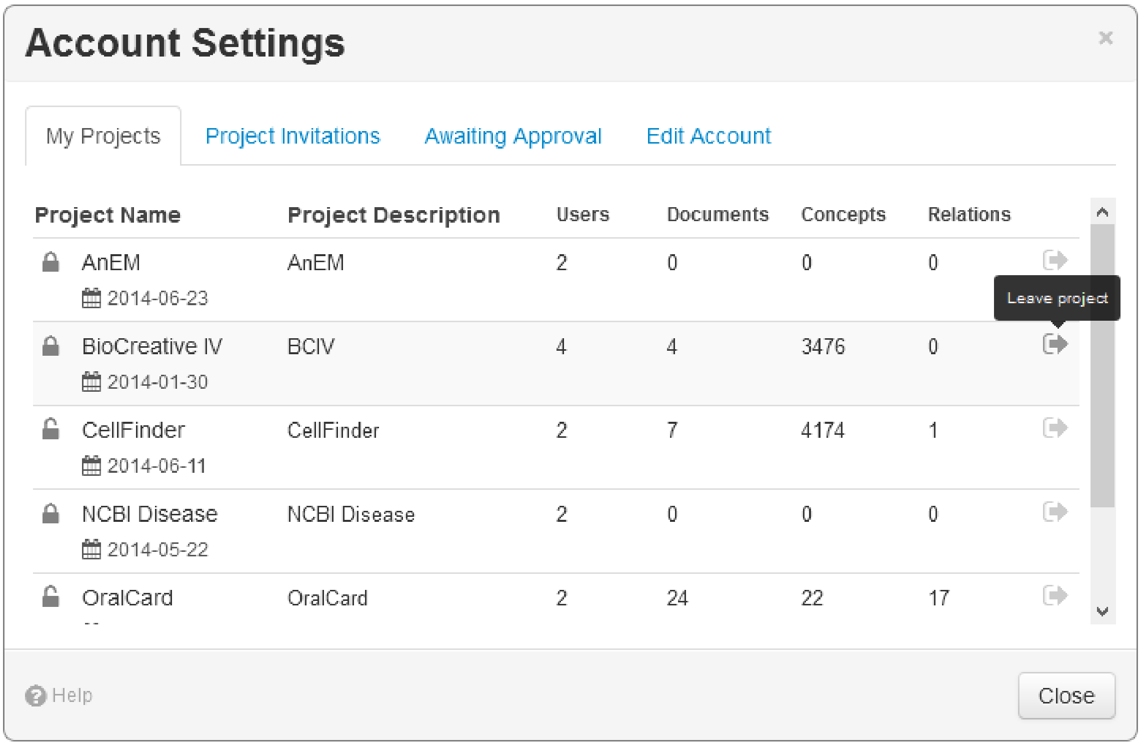
CHEMMER *Chemmer*	2013-09-20	2519	Request invite to project
BioCreative IV *BCIV*	2014-01-30	50	
CellFinder *CellFinder*	2014-06-11	10	You already have access to this project
CRAFT *Colorado Richly Annotated Full Text Corpus*	2014-06-23	67	

Figure 24: Visualization of advanced search window.

5.2.2. User account settings

As an “annotation-as-a-service” platform, Egas allow users to have multiple projects associated with their accounts. In the user account settings window (Figure 25) users can manage all project permissions and account options. In this window users are able to see:

- List of projects associated with the user;
- List of invitations to join other projects;
- List of project requests made by the user that are waiting for approval;
- Users’ account information.



Project Name	Project Description	Users	Documents	Concepts	Relations
AnEM 2014-06-23	AnEM	2	0	0	0
BioCreative IV 2014-01-30	BCIV	4	4	3476	0
CellFinder 2014-06-11	CellFinder	2	7	4174	1
NCBI Disease 2014-05-22	NCBI Disease	2	0	0	0
OralCard --	OralCard	2	24	22	17

Figure 25: Account settings window.

Thereby, users can also leave a certain project, accept or reject invitations to other projects and edit account information, namely email address, name and password. On the other hand, users can create new projects at any time by clicking in the "New Project" button available in the user account menu (Figure 26).

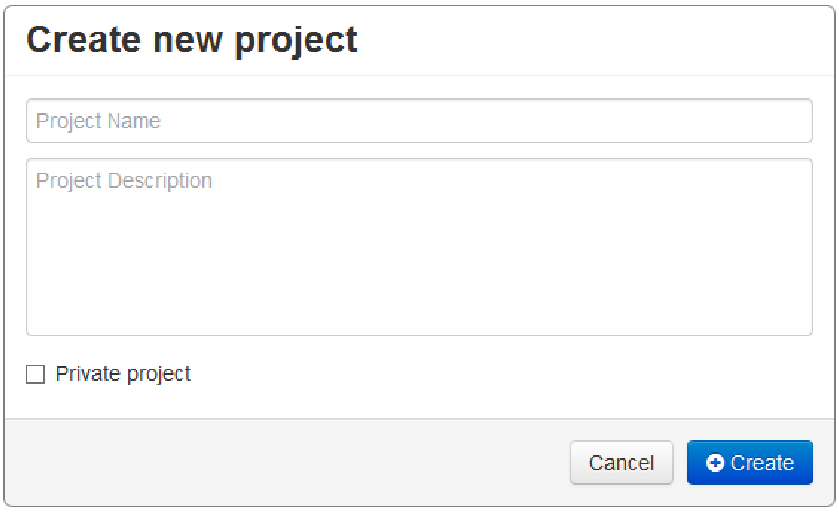


Figure 26: Create new project form. After creating a new project, users can easily configure it using the project wizard.

5.2.3. Project management

In the project management window, project managers can configure essential project characteristics, namely annotation guidelines, manage users, target concept and relation annotations and access the project statistics regarding to the annotation process.

This way, Figure 27 presents the project panel, where administrators can provide de annotation guidelines to the curators through inline text information or by attaching documents in standard formats. Additionally, project managers can choose to show or hide the curation time from the curators while they are annotating documents.

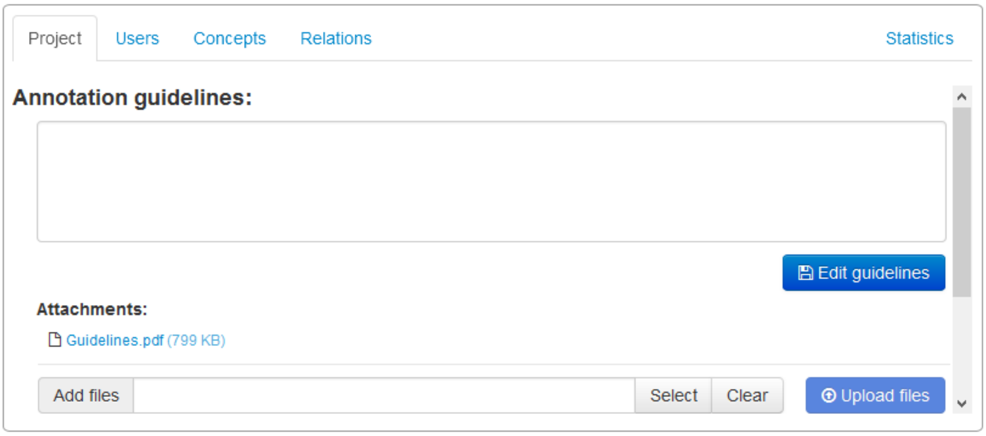
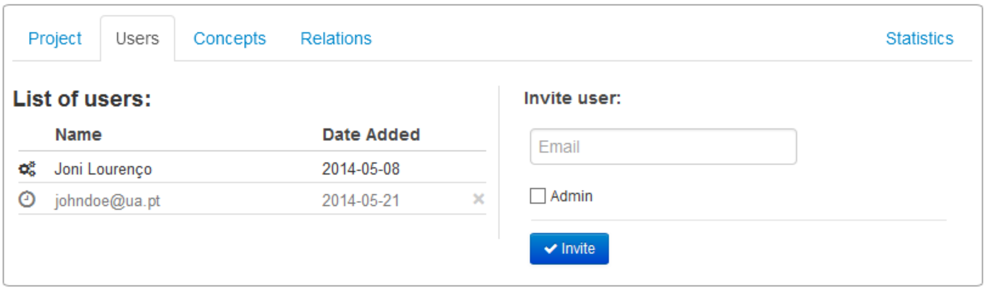


Figure 27: Annotation guidelines in project administrator window.

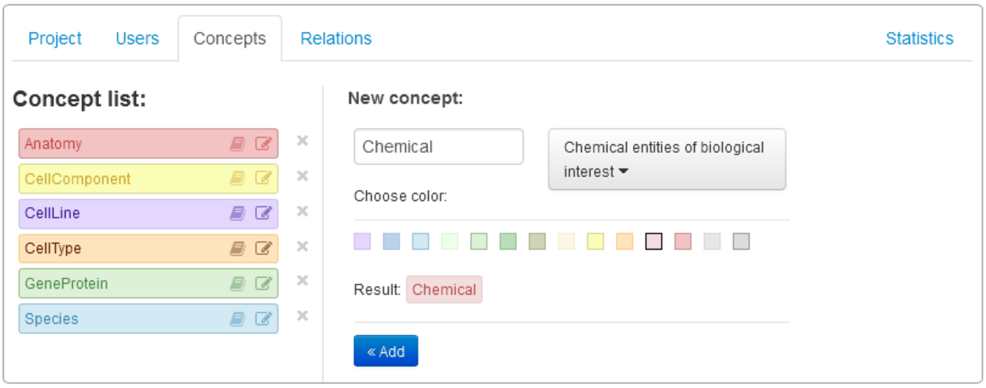
Furthermore, in the user management panel (Figure 28), project managers can invite and remove (✖) users from project. Additionally, this panel also allows managing project administrators (⚙) and pending issued invites (⊙).



	Name	Date Added	
⚙	Joni Lourenço	2014-05-08	
⊙	johndoe@ua.pt	2014-05-21	×

Figure 28: User management in project administration window.

Concepts management (Figure 29) allows administrators to specify the target concepts of the project. Project managers can add new concepts, edit (✎) and/or remove (✖) existing target concepts. When adding new concepts, project managers can also select a knowledge base in order to perform concept normalization. Additionally, the dropdown list containing the available ontologies provides a *typeahead* search box to help users find the desired ontology.

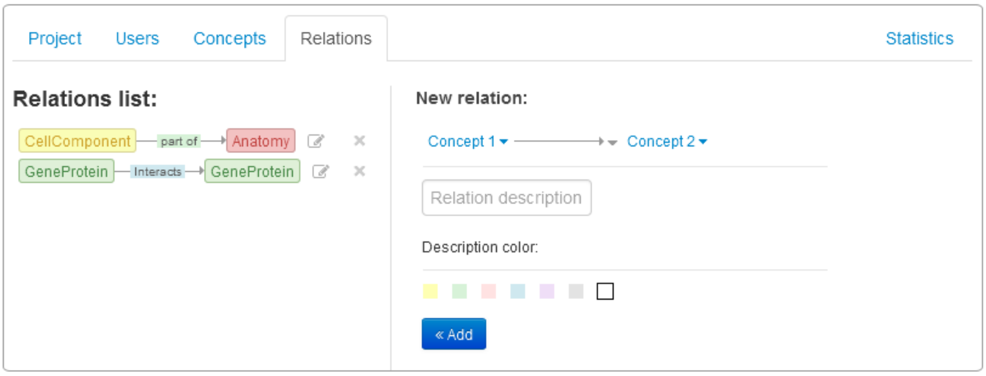


Anatomy
CellComponent
CellLine
CellType
GeneProtein
Species

Chemical
Chemical entities of biological interest
Result: Chemical

Figure 29: Management of target concepts to annotate.

Following the same approach, relations management (Figure 30) allows administrators to specify the target relations of the project.



CellComponent —part of→ Anatomy
GeneProtein —Interacts→ GeneProtein

Relation description

Figure 30: Management of target relations to annotate.

Finally, the statistics panel (Figure 31) allows administrators to collect detailed information regarding the annotation process per article and user, namely curation time and annotated concepts and relations. Egas also allows exporting collected statistics for further analysis.

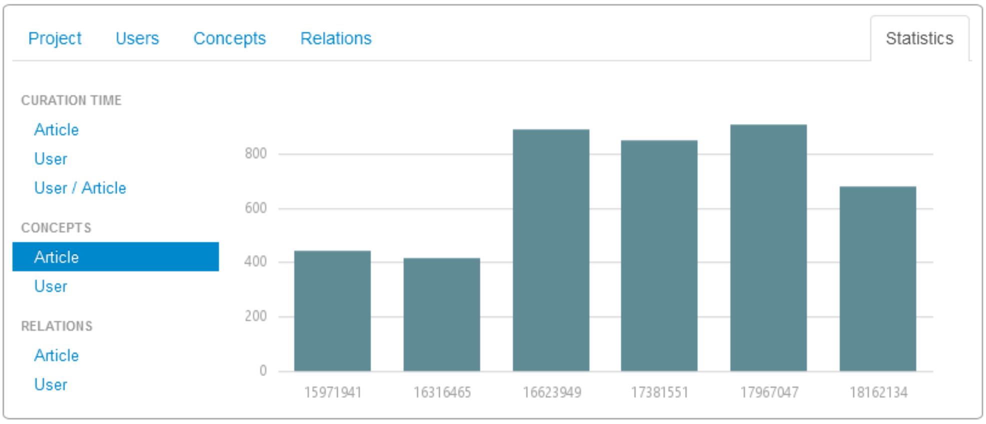


Figure 31: Project statistics in project administration window.

5.2.4. Concept and relation annotation

Information annotation is the central feature of Egas, which provides easy and interactive annotation of concepts and relations. This way, curators can annotate chunks of text according to the target concepts and also define relations between those same concepts.

Add concept annotation

In order to add concept annotations, users must be registered in Egas and make part of the project. Thereby, concept annotation is performed in three simple steps (Figure 32):

1) Select chunk of text;
2) Select concept type from the available target concepts;
3) The concept annotation is created and highlighted with the respective concept color.

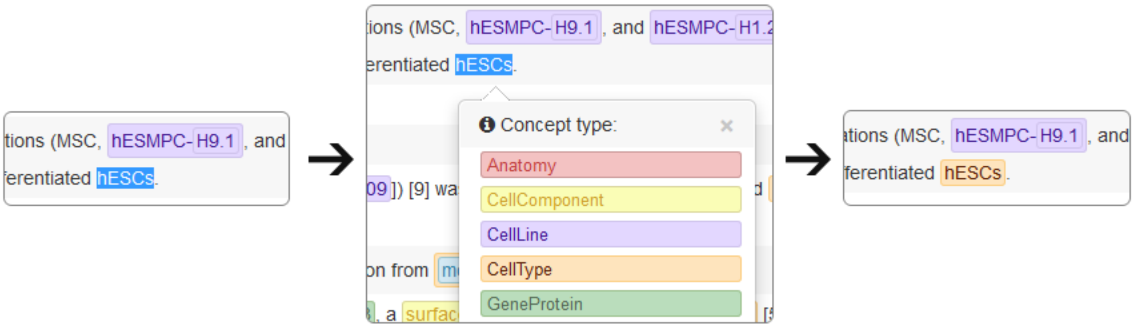
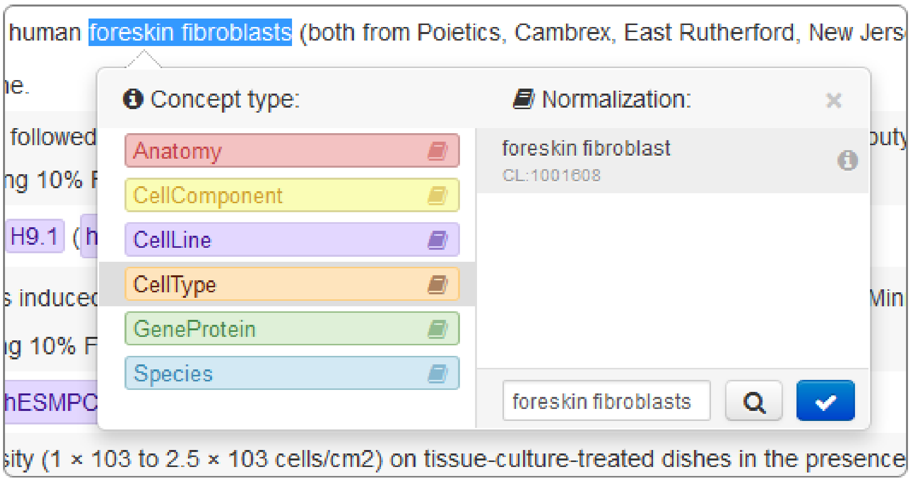


Figure 32: User interface interaction steps to add a new concept annotation.

On the other hand, if the selected concept type is associated with a certain knowledge base, another step is required in order to perform concept normalization. Thus, when adding a new concept, curators must select the desired identifier from the suggested list, retrieved by a simple text query to the respective Solr core (Figure 33). Furthermore, if the available suggestions do not fit the desired identifier, users can search the knowledge base, by keyword, for the correct identifier. Nevertheless, even though some concept types are associated with some ontologies, curators are free to insert new concepts without normalization information.



human foreskin fibroblasts (both from Poietics, Cambrex, East Rutherford, New Jers

sity (1 × 103 to 2.5 × 103 cells/cm2) on tissue-culture-treated dishes in the presence

Figure 33: User interface of adding a new concept annotation and associate it to a unique identifier.

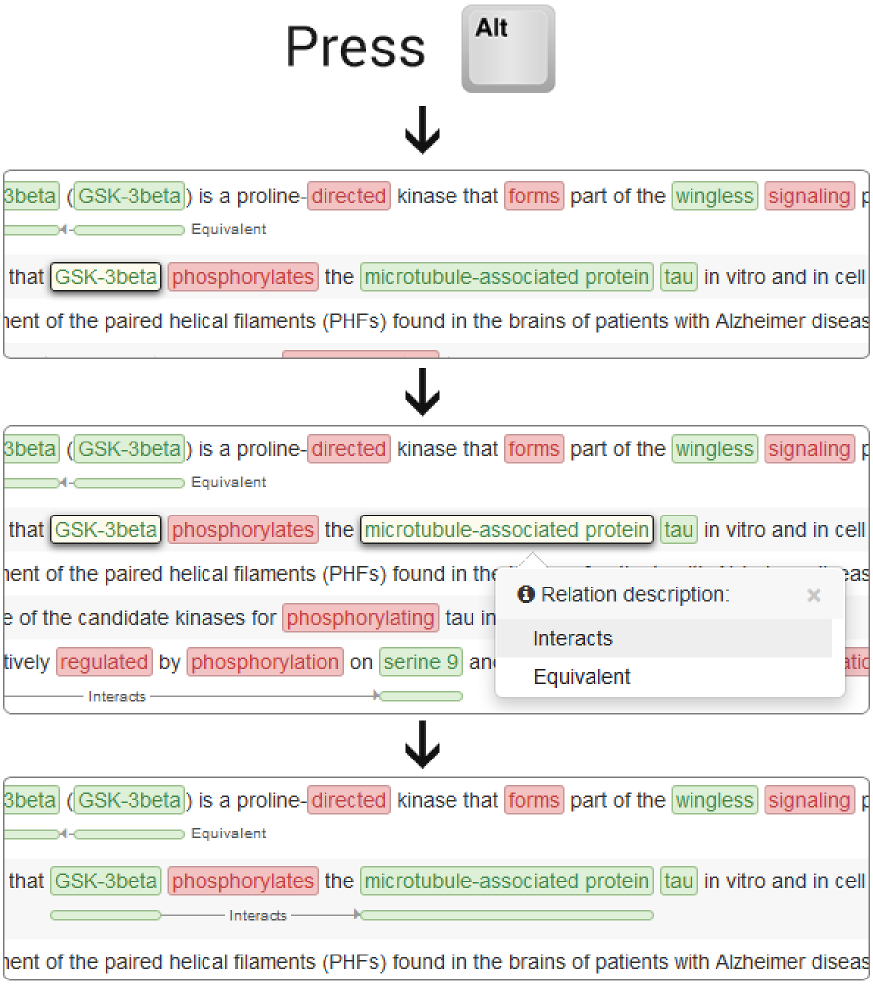
Add relations

Creating relations between concepts is also an easy-to-use feature of Egas. Thus, adding concept relation annotations is performed in four simple steps (Figure 34):

1) Keep pressing the "Alt" keyboard key;
2) Click on the first concept name;
3) Click on the second concept name and, if more than one relation type between these two concept types is defined, select the relation type from the available list;
4) The relation annotation is created and highlighted below the sentence with the respective relation direction and description.

In some cases project managers define more than one relation type between two concepts. For instance, an annotation tasks could require relations between "Proteins" with different descriptions, such as "Interacts" and "Not interacts". In this case, when selecting the second concept of the relation, the system provides a list of the available relation types in order for the users to choose the desired one. On the other hand, if only one relation type fits the two selected concepts, the relation annotation is immediately created when the user selects the second concept.

Additionally, if curators try to create relations between concepts that were not defined by the project managers, the system warns that the desired relation is not defined in the target relations to annotate.



3beta (GSK-3beta) is a proline-directed kinase that forms part of the wingless signaling p

that GSK-3beta phosphorylates the microtubule-associated protein tau in vitro and in cell

nent of the paired helical filaments (PHFs) found in the brains of patients with Alzheimer diseas

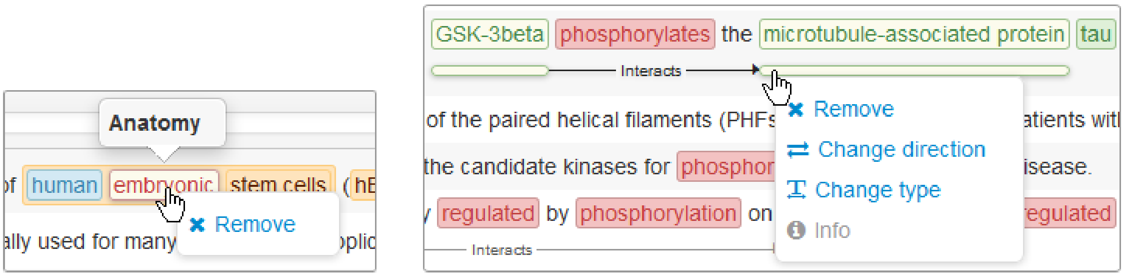
e of the candidate kinases for phosphorylating tau in

tively regulated by phosphorylation on serine 9 an

Figure 34: User interface interaction steps to add a new relation.

Annotation options

Finally, removing existing concept or relation annotations is also possible by right-clicking on the annotation and choosing the option remove. Furthermore, curators can change the relation type and direction in the available context menu of relation annotations (Figure 35).



of human embryonic stem cells (hE

ally used for many applic

GSK-3beta phosphorylates the microtubule-associated protein tau

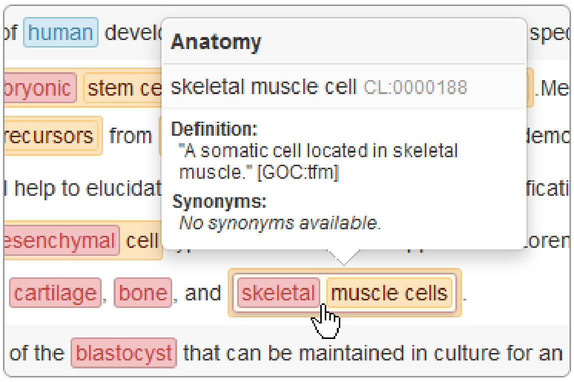
of the paired helical filaments (PHFs atients wit

the candidate kinases for phosphor isease.

regulated by phosphorylation on egulated

Figure 35: Context menu of document annotations.

Moreover, by moving the cursor over concept annotations users are able to see more information about the concept such as the concept type and, if is defined, the normalization information (Figure 36).



Anatomy

skeletal muscle cell CL:0000188

Definition:
"A somatic cell located in skeletal muscle." [GOC:tfm]

Synonyms:
No synonyms available.

of human devel
spec
bryonic stem ce
Me
recursors from
lemo
help to elucidat
ficati
esenchymal cell
oren
cartilage, bone, and skeletal muscle cells.
of the blastocyst that can be maintained in culture for an

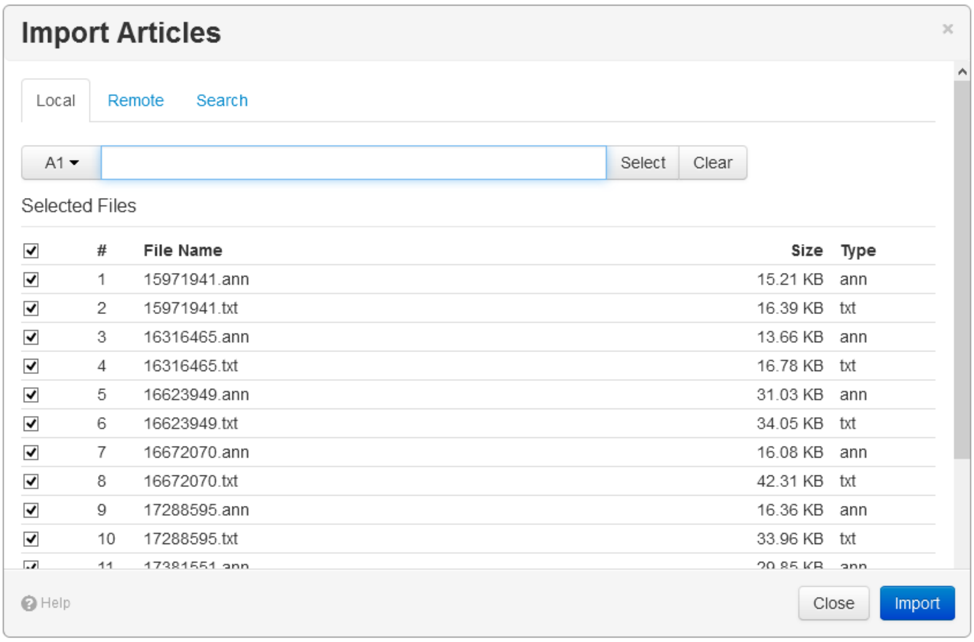
Figure 36: Mouse over popup showing normalization information of a concept annotation.

5.2.5. Import and export documents

In order to annotate biomedical texts, users need to import documents to the currently selected project. Thus, Egas's import window, allows users to import documents from both local and remote servers. Egas currently supports both PubMed and PubMed Central services, in order to import abstracts and full-text documents, respectively. Thereby, documents can be imported in three different ways:

1) Local: allows users to import documents stored on their computer. This feature supports three formats: raw text, A1 and BioC (Figure 37);
2) Remote: allows users to add documents from the remote services through a list of unique identifiers (Figure 38);
3) Search: allows users to import documents by searching remote services that already have publicly available literature indexed (Figure 39).

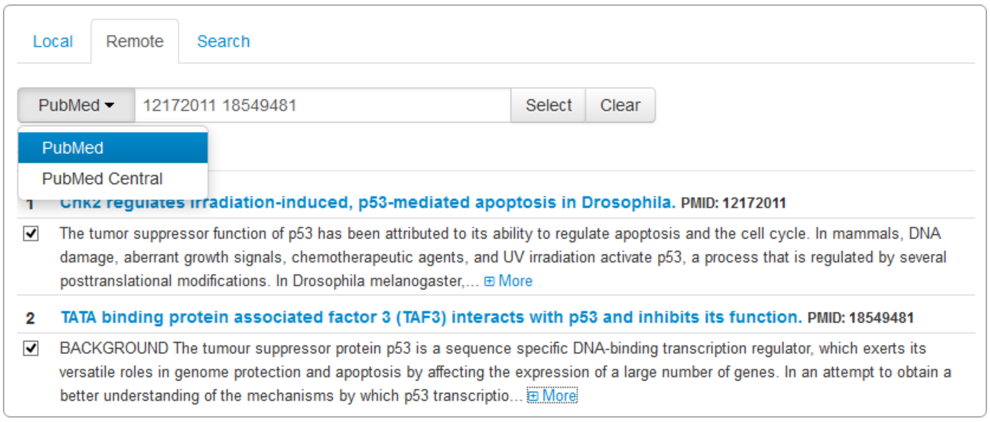
User queries are executed directly on the remote services, allowing logic operators such as "AND" and "OR", as well as MeSH type queries. After submitting the query, Egas presents a ranked list of documents, and allows the users to select the documents they want to import.



Selected Files

☑	#	File Name	Size	Type
☑	1	15971941.ann	15.21 KB	ann
☑	2	15971941.txt	16.39 KB	txt
☑	3	16316465.ann	13.66 KB	ann
☑	4	16316465.txt	16.78 KB	txt
☑	5	16623949.ann	31.03 KB	ann
☑	6	16623949.txt	34.05 KB	txt
☑	7	16672070.ann	16.08 KB	ann
☑	8	16672070.txt	42.31 KB	txt
☑	9	17288595.ann	16.36 KB	ann
☑	10	17288595.txt	33.96 KB	txt
☑	11	17381551.ann	20.85 KB	ann

Figure 37: Import documents from local source.



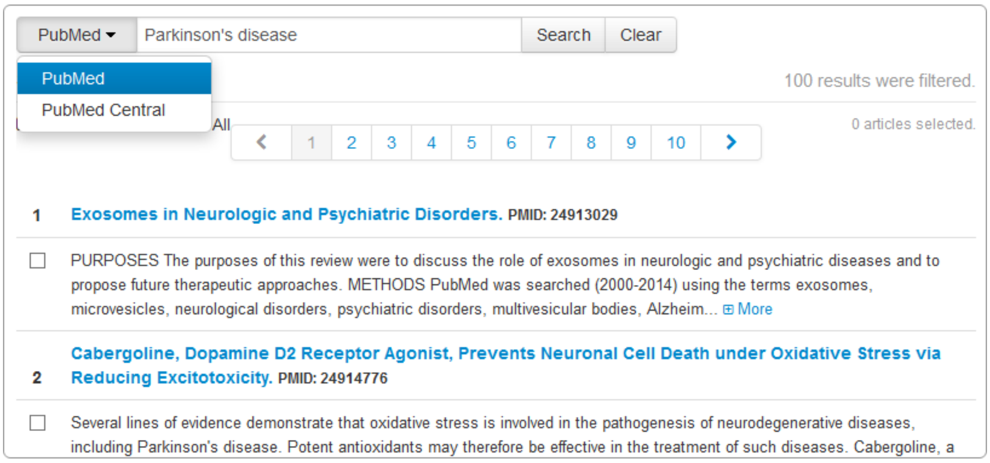
1 Cnk2 regulates irradiation-induced, p53-mediated apoptosis in Drosophila. PMID: 12172011

☑ The tumor suppressor function of p53 has been attributed to its ability to regulate apoptosis and the cell cycle. In mammals, DNA damage, aberrant growth signals, chemotherapeutic agents, and UV irradiation activate p53, a process that is regulated by several posttranslational modifications. In Drosophila melanogaster,... ⊞ More

2 TATA binding protein associated factor 3 (TAF3) interacts with p53 and inhibits its function. PMID: 18549481

☑ BACKGROUND The tumour suppressor protein p53 is a sequence specific DNA-binding transcription regulator, which exerts its versatile roles in genome protection and apoptosis by affecting the expression of a large number of genes. In an attempt to obtain a better understanding of the mechanisms by which p53 transcriptio... ⊞ More

Figure 38: Import documents from remote sources by PubMed and PubMed Central identifiers.



100 results were filtered.

0 articles selected.

1 Exosomes in Neurologic and Psychiatric Disorders. PMID: 24913029

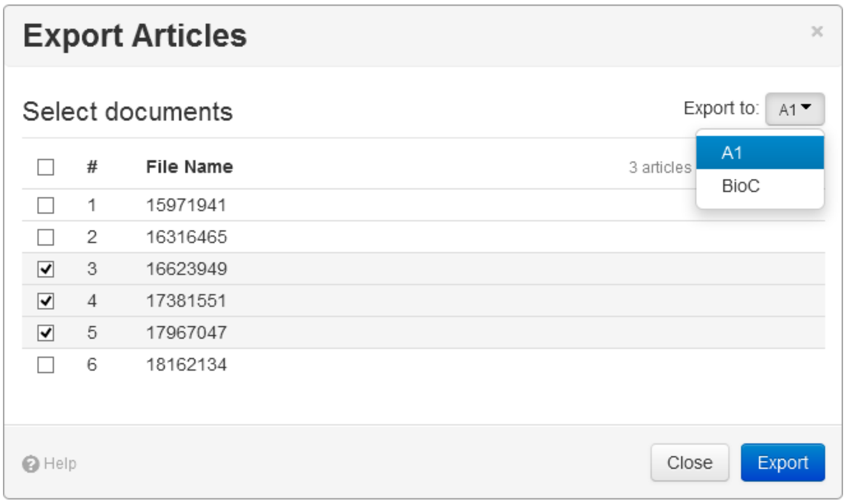
□ PURPOSES The purposes of this review were to discuss the role of exosomes in neurologic and psychiatric diseases and to propose future therapeutic approaches. METHODS PubMed was searched (2000-2014) using the terms exosomes, microvesicles, neurological disorders, psychiatric disorders, multivesicular bodies, Alzheim... ⊞ More

2 Cabergoline, Dopamine D2 Receptor Agonist, Prevents Neuronal Cell Death under Oxidative Stress via Reducing Excitotoxicity. PMID: 24914776

□ Several lines of evidence demonstrate that oxidative stress is involved in the pathogenesis of neurodegenerative diseases, including Parkinson's disease. Potent antioxidants may therefore be effective in the treatment of such diseases. Cabergoline, a

Figure 39: Search and import documents from PubMed and PubMed Central.

On the other hand, Egas allows users to store documents and respective annotations on their local machines, supporting the same three formats: raw text, A1 and BioC. Through the provided interface (Figure 40) users can select the documents to be exported, which are provided in a single compressed file.



Export Articles

Select documents

Export to: A1

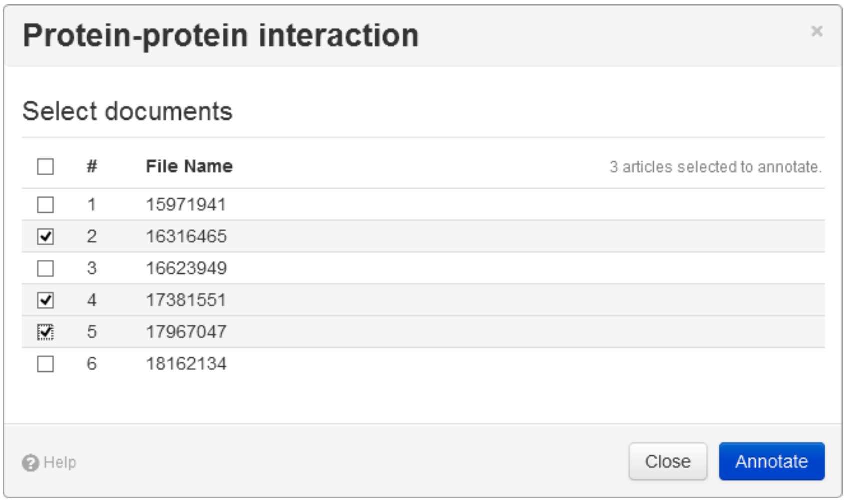
3 articles

□	#	File Name
□	1	15971941
□	2	16316465
☑	3	16623949
☑	4	17381551
☑	5	17967047
□	6	18162134

Figure 40: User interface to export project documents and respective annotations.

5.2.6. Automatic annotation services

Egas user interface also allows calling automatic annotation services for specific documents. This interface was developed to be as flexible and adaptable as possible, in order to support services with different characteristics. Thus, Egas only requires the user to indicate the documents that should be annotated by the service (Figure 41). Afterwards, the respective documents' annotations are loaded to Egas and presented in the document viewer.



Protein-protein interaction

Select documents

3 articles selected to annotate.

□	#	File Name
□	1	15971941
☑	2	16316465
□	3	16623949
☑	4	17381551
☑	5	17967047
□	6	18162134

Figure 41: Integrated automatic annotation services. Users only need to indicate the documents that should be annotated by the service.

5.2.7. Real-time collaborative features

Finally, as part of the Egas workspace, it is also possible to enable innovative real-time collaboration features. That way, Egas provides instant feedback of users' interactions within a document, such as adding, removing and/or changing concept and relation annotations (Figure 42). Furthermore, multiple users can change a document at the same time, showing exactly who changed what.

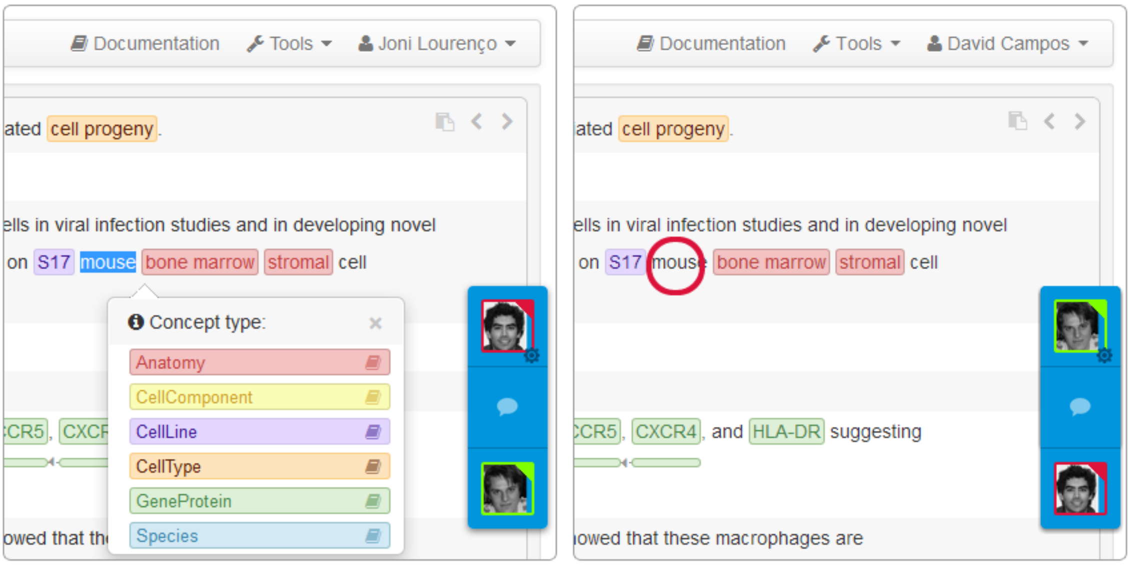
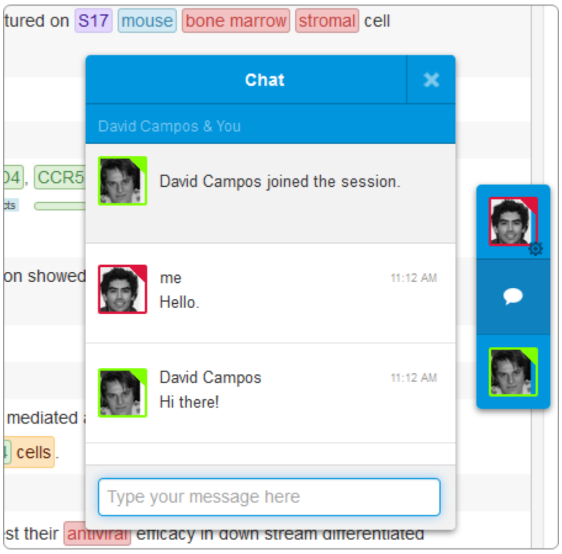

Documentation
Tools
Joni Lourenço
ated cell progeny.
ells in viral infection studies and in developing novel
on S17 mouse bone marrow stromal cell
Concept type:
Anatomy
CellComponent
CellLine
CellType
GeneProtein
Species
CR5, CXCR
owed that th
Documentation
Tools
David Campos
ated cell progeny.
ells in viral infection studies and in developing novel
on S17 mouse bone marrow stromal cell
CR5, CXCR4, and HLA-DR suggesting
owed that these macrophages are

Figure 42: Egas's real-time collaborative features, providing instant feedback when other users are annotating documents.

A project chat is also available (Figure 43), allowing users to discuss details of the annotation task. Moreover, the system also provides feedback indicating where on the screen, remote users clicked.


tured on S17 mouse bone marrow stromal cell
Chat
David Campos & You
04, CCR5
on showed
mediated
cells.
st their antiviral efficacy in down stream differentiated
Type your message here

David Campos joined the session.

me 11:12 AM
Hello.

David Campos 11:12 AM
Hi there!

Figure 43: Egas's collaborative chat window implemented by taking advantage of TogetherJS library.

5.3. Implementation details

In this section specific details regarding the implementation of Egas will be provided, namely the document representation algorithm, automatic annotation services details and some of the provided features.

5.3.1. Document parsing and representation

One of the most important features of interactive mining solutions is the inline representation of annotations. Likewise, an intuitive representation of the annotated document enables a context-based interaction with the generated information and significantly improves the curator job as well as his understanding of the working document. However, as far as we know, existing solutions for interactive mining take advantage of Scalable Vector Graphics (SVG) to display inline annotations of concepts and relations. Nevertheless, this technique presents some issues regarding application performance, since the representation of large documents containing thousands of annotations may take a considerable amount of time to render and display on the browser window. Likewise, this limitation affects substantially the performance of the application making the curator's job harder unlike what would be expected.

To address this issue and improve document loading times, we decided to use only HTML, CSS and JavaScript technologies. Even though they are less flexible in terms of representation capabilities, they are extremely faster regarding to document rendering and performance, which allows us to completely display large documents with thousands of annotations in just a few seconds. However, the representation of concept and relation annotations using these technologies requires the application of some ingenious techniques.

The document representation process (Figure 44) begins by retrieving the necessary data from the server, namely document text (divided in sentences) and respective concept and relation annotations. Additionally, before document representation, the application resets all document related data and configures the workspace accordingly to the selected project and document. On the other hand, if the collaborative mode is activated, TogetherJS properties are configured in order to update the room and active curators. To simplify the annotation process, the document text is presented by sentence, making it more focused in the main message contained on each sentence. The sentence split process is performed on the server-side using Lingpipe[35], through a model trained on documents from the biomedical domain.

[35] http://alias-i.com/lingpipe/

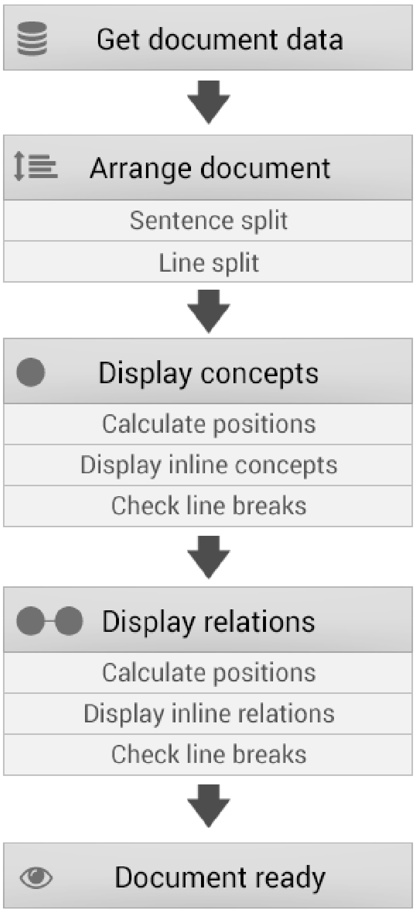
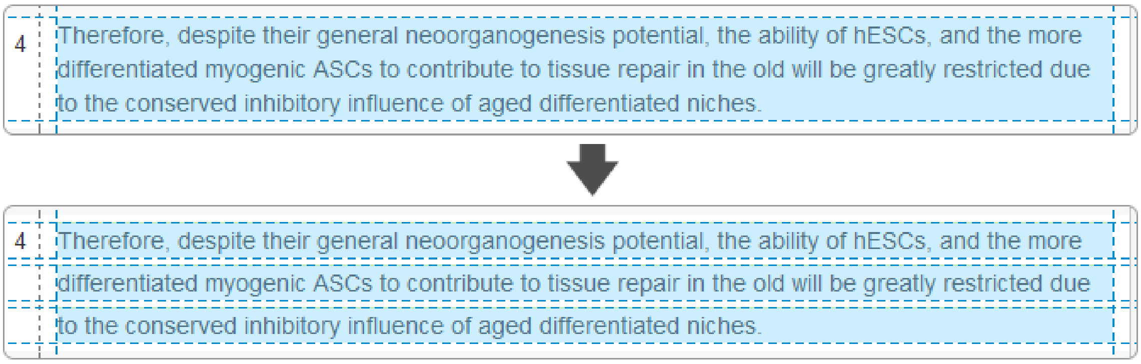


Figure 44: General overview of required procedures to load a document and respective annotations in Egas's workspace.

Depending on sentence length and browser window width, some sentences may be represented in more the one line. Even though browsers automatically break lines of text according to containing element width, we need the space between the sentences lines to represent relation annotations. Thus, it is fundamental to split every sentence into multiple containers, one for each line. Thereby, this task is a very important step in order to represent the document annotations properly. Such goal is achieved by adding word by word to a *div* element until the maximum width per line is reached. When this happens, the *div*'s height changes and the system breaks the text in the previous position, and adds the remaining words into a new line. This process continues until there is no more text on the corresponding sentence. In the end, every line is inside of a respective container allowing us to proceed to the next step of the document rendering algorithm. Meanwhile, the information regarding the character positions of each sentence is stored internally.

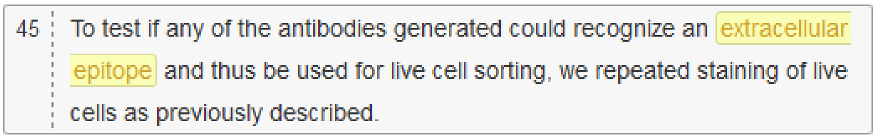


4 Therefore, despite their general neoorganogenesis potential, the ability of hESCs, and the more differentiated myogenic ASCs to contribute to tissue repair in the old will be greatly restricted due to the conserved inhibitory influence of aged differentiated niches.

4 Therefore, despite their general neoorganogenesis potential, the ability of hESCs, and the more
differentiated myogenic ASCs to contribute to tissue repair in the old will be greatly restricted due
to the conserved inhibitory influence of aged differentiated niches.

Figure 45: Representation of line splitting process. The top of the figure represents the *div* element containing the entire sentence. In the end of this process, as we can see in the bottom of this figure, each line is represented inside an individual *div* element.

After arranging the documents' text, we are ready to add the concept annotations. The list of concepts previously retrieved from the server, contains the unique identifier, concept type, and start and end positions of every concept of the selected document. This way, we start by analyzing each sentence and open a *span* element tag with the respective concept identifier on each start position and add the respective closing tag when the end position is reached. This process continues until we reach the end of each line, where we write the annotated sentence into the page. Nevertheless, when an annotation starts in one line but ends in the next one, special open and close span tags are added to give the idea of a continuous annotation (Figure 46).

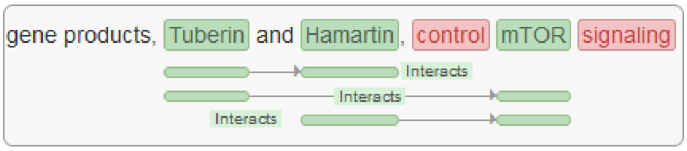


45 To test if any of the antibodies generated could recognize an extracellular
epitope and thus be used for live cell sorting, we repeated staining of live
cells as previously described.

Figure 46: Display of concept annotations that start and end in different lines.

In the end of this process, all concept annotations have a respective inline representation in the document. On the other hand, since the list of concepts is already sorted by starting position when retrieved from the server and as we only manipulate the document HTML when every sentence is ready, this process takes, most of times, less than one second to be performed, even when we consider large documents with several thousands of annotations.

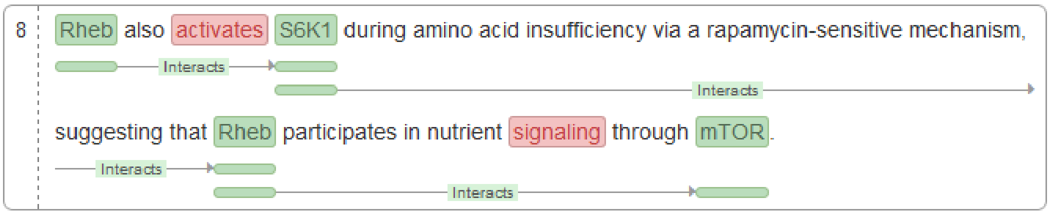
Finally, we need to add the relation annotations to the document. Like the previous steps, this process is performed sequentially sentence by sentence. Then, each sentence, the respective relations are displayed in sequential order by adding a box below each concept, and a line connecting the two boxes with a corresponding arrow pointing the direction. Likewise, the label with the relation description is placed where space is available, considering the space between concepts box, or in the left or right sides of the relation, by this priority (Figure 47).



gene products, Tuberin and Hamartin, control mTOR signaling

Figure 47: Different representation of the relation description, placing the label where space is available.

Furthermore, the display of multiple relations in the same line is performed by placing them vertically, aligned with the respective concepts boxes. However, when the relation traverses multiple text lines, the line connects the two concepts by traversing the various text lines, and arrows and labels are repeated on each line to keep context (Figure 48). At this point the representation process is finished and the user has access to all annotations inline the document.



8 Rheb also activates S6K1 during amino acid insufficiency via a rapamycin-sensitive mechanism,

suggesting that Rheb participates in nutrient signaling through mTOR.

Figure 48: Representation of a relation that begins in one line and ends in the next one.

Despite the most ingenious and time-consuming part is completed, there are a few more actions that we need to be aware of in order to keep the correct representation of documents and respective annotations. Likewise, performing actions such as adding or removing concept and relation annotations or even resizing the browser window, can affect the document visualization and, thereby, should be handled correctly. Thus, after adding or removing concept annotations from the document, the affected line is completely redesigned so that the remaining items match the new positions of the relation boxes and respective concepts. On the other hand, since every concept box adds a few pixels to the corresponding line, creating a new relation can generate a line break, which needs to be handled in order to keep the correct representation of the entire line. However, creating new relation annotations only require the adjustment of the space between lines and does not affect the inline concept visualization.

On the other hand, every time a user resizes the browser window, the document representation must be redesigned according to the new width of the page. However, since the required data is already loaded in memory, this process is considerably faster. Furthermore, to improve the application performance, the redesign algorithm is only triggered when the window resize is completed and only if the browser width was affected. Additionally, changing the browsers' text size will also trigger the representation redesign.

5.3.2. Import and export documents

Like it was previously described, Egas supports importing articles and respective annotations from local and remote servers. Moreover, importing documents' annotations is only possible in local import since remote servers only provide the documents' text. Thus, when importing local documents into a specific project, if document annotations are already available, the application automatically imports them to the database and presents them in the document viewer. Thereby, in order to import documents and annotations presented in BioC and A1 formats to the database, methods to correctly parse the files and import the information to Egas's database were developed. Likewise, the developed algorithms, firstly read the input files and store the respective document text in the database. Secondly, if the imported documents have annotations to import to the database, each concept annotation is parsed and stored. Thereafter, the relation annotations are parsed, associated with the previous stored concepts and imported to the database. Furthermore, in order to improve performance and avoid database overload, all the data insertions are performed in batch mode.

Importing documents from remote sources is performed using the respective web-services of PubMed and PubMed Central. These services were integrated with Egas's provided services to centralize and secure all data connections. The remote services already support retrieving specific documents by unique identifier or by search query. Thus, we integrated PubMed services through the E-utility Simple Object Access Protocol (SOAP) web service [77] and PubMed Central using the Open Access (OA) REST web services[36]. In the end, as expected, users are able to search and select PubMed and PubMed Central documents and import them directly into their project workspace.

On the other hand, Egas also provides features to export annotated documents in different formats so users can further analyze and share the generated concept and relation annotations. Likewise, Egas currently supports exporting documents in two different formats: A1 and BioC. Thus, users can select documents from the working project and export them in a single compressed file. In case of BioC format, all the exported documents are merged in a single XML file.

The process of parsing and writing documents in BioC format is performed by taking advantage of the publicly available BioC Java library[37].

5.3.3. Annotation services

To deliver a high-level solution and improve the curators' results, Egas's provides a set of automatic annotation services. Thus, users can call an automatic annotation service

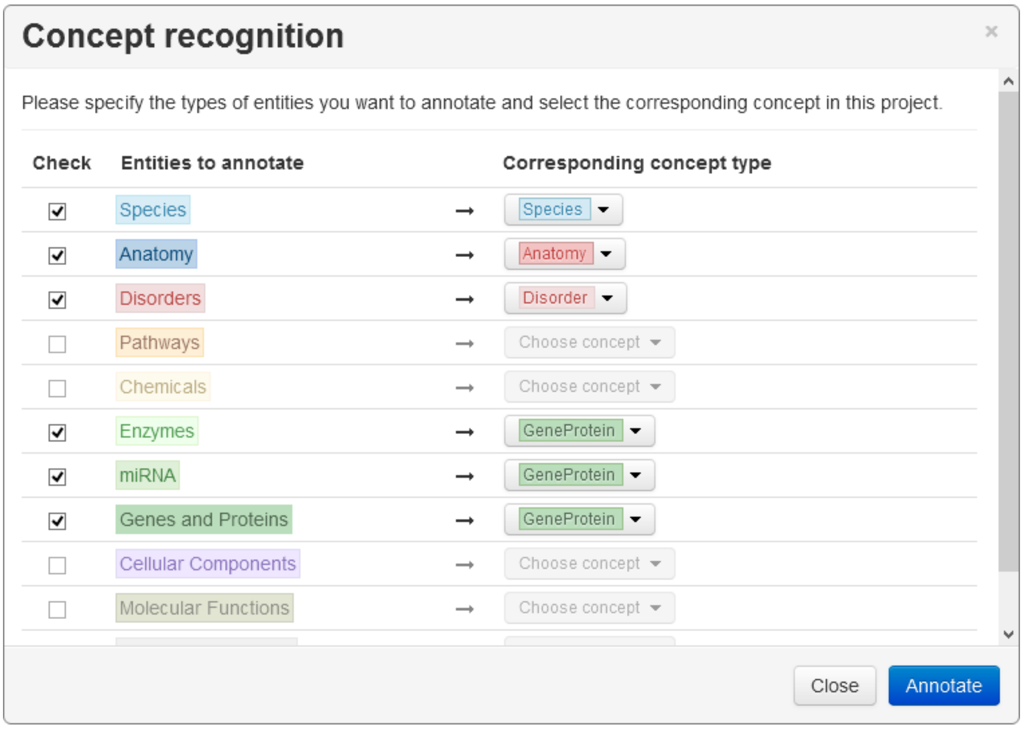
[36] http://www.ncbi.nlm.nih.gov/pmc/tools/oa-service

[37] http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/BioC

performing identification of specific concepts and/or relations using state-of-the-art algorithms and posteriorly manually correct the provided annotations and/or add missing ones. In fact, several studies have shown that the curation times have been improved when curators use automatic solutions to assist their tasks [8, 9]. Therefore, these services intend to considerably decrease the amount of time spent in the manual curation process.

Egas supports annotation services through a flexible and simple REST web service interface. Thereby, the annotation services have to accept text as input and provide the respective annotations following the A1 or BioC formats as output. It is straightforward to add new services to identify different concepts and/or relations to the application, which make it ready to integrate more services at any time.

Furthermore, two different automatic annotation services were implemented in Egas's interface, one for concept recognition and one for protein-protein interaction mining. Thereby, the concept identification service takes advantage of the BeCAS [78] REST API to provide annotations of genes and proteins, species, anatomical concepts, miRNAs, enzymes, chemicals, drugs, diseases, metabolic pathways, cellular components, biological processes and molecular functions. This way, in Egas's user interface, after selecting the documents to perform automatic annotation of concepts, users must associate the target concepts of the selected project to the ones provided by the BeCAS service in order to cast the automatic annotated concept names to the ones defined by the project manager (Figure 49).



Concept recognition

Please specify the types of entities you want to annotate and select the corresponding concept in this project.

Check	Entities to annotate		Corresponding concept type
☑	Species	→	Species
☑	Anatomy	→	Anatomy
☑	Disorders	→	Disorder
□	Pathways	→	Choose concept
□	Chemicals	→	Choose concept
☑	Enzymes	→	GeneProtein
☑	miRNA	→	GeneProtein
☑	Genes and Proteins	→	GeneProtein
□	Cellular Components	→	Choose concept
□	Molecular Functions	→	Choose concept

Figure 49: User interface to associate the target concepts of the selected project to the ones provided by the BeCAS annotation service.

Finally, the PPIs automatic service provides the following annotations: protein concept recognition, relations between proteins, relations marking equivalent protein mentions (e.g. acronyms and long forms), and active words that may indicate the presence of PPIs. The integrated service is implemented on top of Neji [79], using Gimli [80] to perform ML-based protein name recognition. Moreover, BioThesaurus is used to normalize recognized names, through the application of a prioritized dictionary matching strategy.

5.3.4. Web-service security and role-based access control

Security and authentication are also an important and relevant factor that we need be aware of. By delivering a web-based platform, we need to assure that all the stored data and every connection between the server and the client are properly authenticated and secured. On the other hand, in our application, users have different levels of access, such as curator, project manager or system administrator, as well as different users may have access to different projects. Thus, in order to fulfil these requirements, we have designed and implemented a role-based access control to all of the provided services. This way, we make sure that every method called from the client-side of the application is executed only if the right user has requested the information, preventing security breaches and assuring that the stored data is consistent and secure.

As previously described, the web service was developed in Java using the Jersey RESTful framework. In order to implement a flexible and extensible security platform, we took advantage of Spring Security framework, which provides a powerful and highly extensible authentication and access-control framework. Thus, we carefully analyzed the application requirements regarding to access levels and defined the following roles (Table 5.1):

- Logged User: this role is assigned to all users after they log in Egas web interface. It is used to distinguish logged users from visitors.
- Guest User: assigned to all visitors in order to get access to public stored data.
- Curator: represents a user that is associated with a project, thus, has curator access. This role is assigned according to users' access to a certain project, i.e., a particular user only has curator access in projects that he is associated with.
- Project Manager: like the curator role, is assigned to users that have manager permissions in a certain project.
- System Administrator: this role represents all the Egas's system administrators. Likewise, they have full access to the application, thus, to all methods delivered by the web service.

Action	Guest User	Logged User	Project Curator	Project Admin	System Admin
Create project	✗	✓	✓	✓	✓
Access public projects	✓	✓	✓	✓	✓
Access private projects	✗	✓	✓	✓	✓
Add/remove concept annotations	✗	✓	✓	✓	✓
Add/remove relation annotations	✗	✓	✓	✓	✓
Use automatic annotation services	✗	✓	✓	✓	✓
Import articles	✗	✓	✓	✓	✓
Export articles	✓	✓	✓	✓	✓
Delete project articles	✗	✗	✗	✓	✓
Manage project settings	✗	✗	✗	✓	✓
Manage project users	✗	✗	✗	✓	✓
Delete projects	✗	✗	✗	✗	✓
Manage global system settings	✗	✗	✗	✗	✓

Table 5.1: Different roles and respective permissions in Egas user interface.

Therefore, in order to assure proper authentication and authorization of the services, we assign the corresponding role/roles to each method of the provided resources. Moreover, after loading Egas's web application, every user is assigned with a session-id which is sent in the header information of each one of the further requests made to the web service. On the other hand, the server stores the information regarding to active users, such as user-id, session-id and respective assigned roles and project permissions. Thus, in every incoming request, the session-id authenticates the respective user in order to check his permissions. Thereafter, if the user is in one of the roles allowed of the requested method, the request is successful, otherwise the request is denied.

As previously described, to properly authenticate every request, the server stores the information of each active session and respective user. To do so, we implemented a SessionRepository and a UserRepository, containing the session and user information's respectively. Thus, whenever a session is created or destroyed, SessionRepository will intercept the generated event in order to maintain an updated list of sessions and respective users. Moreover, now that we already have a way to store all session and user information, we need to create a SecurityContext that will be bound to incoming requests and will decide whether to allow or deny it. SecurityContext is responsible for checking if a User is in one of the allowed roles of the requested method. Accordingly, since we have roles based on project permissions, some methods may require additional operations. For example, a particular user may have manager permissions in project 1 and only curator access to project 2. Likewise, this user is in Project Manager role for project 1 and in Curator role for project 2. This approach makes these roles "flexible" and requires some additional processing. In this case, when accessing methods, for instance, regarding to project administration, SecurityContext checks if user has the respective project id in his project management list, allowing or denying the request according to this verification.

The implemented SecurityContext allows us to accept or deny each request based on the user assigned roles. However, we need to implement a ResourceFilter which will intercept the requests, retrieve the session id in the header as well as project id (if defined), and generate and attach our previous implemented SecurityContext to it. On the other hand, SecurityContext checks the user permissions and decides whether to allow or deny the request.

Finally, in order to trigger our ResourceFilter in every request, we created a RolesAllowedResourceFilterFactory. Thus, during application startup, this factory will create a list of filters for all methods of each of our Resources. In the end, after login, a session is created and assigned to the respective user. In the session information we can find the user roles as well as his project permissions. Afterwards, the implemented ResourceFilter intercepts all incoming requests, retrieves the session id from the header and attaches the SecurityContext to the request. Then, the SecurityContext checks user roles and decides whether to allow or deny the request.

In summary, with the developed role-based access control security system, we achieve a robust and secure web service, providing trustful services and avoiding possible undesired requests. Additionally, the communication between client and server is performed using a secured and encrypted channel using Hypertext Transfer Protocol Secure (HTTPS), in order to guarantee complete protection of exchanged data.

Chapter 6

Results and discussion

This section presents the results of the developed system. Furthermore, Egas was part of an experiment performed to evaluate the solution applicability in a real-life scenario, evaluating the platform impact and users' satisfaction. In the next pages we will describe the respective experiment and analyze the obtained results.

6.1. BioCreative IV experiment

In order to evaluate the applicability and satisfaction of using Egas, we participated in the BioCreative IV interactive task [59], which intended to promote the development of useful text mining tools to fill the gap between biocuration and text-mining research communities, exploring the user-system interactions and hidden requirements. Thus, the task targeted the development of solutions to support the interactive mining and/or triage of scientific documents.

The task organizers, together with a group of expert curators, defined a prioritized list of system requirements that they considered more important to be available in such system. Thereby, the five most important system requirements were:

1) Highlight entities and relationships;
2) Process full text;
3) Allow manual mode for annotation;
4) Ability to edit results;
5) Export curated results in standard formats.

Each participating team developed and submitted their own approach to deal with the complied specifications. Moreover, each team had to propose a biocuration task to apply and test-drive the presented system. Our proposal consisted in the identification and extraction of biomolecular events described over PubMed abstracts related to neuropathological disorders, including PPIs, protein expression and post-translational modifications. To create the corpus for this task, a collection consisting of more than 135 thousand PubMed abstracts was first obtained with the following query:

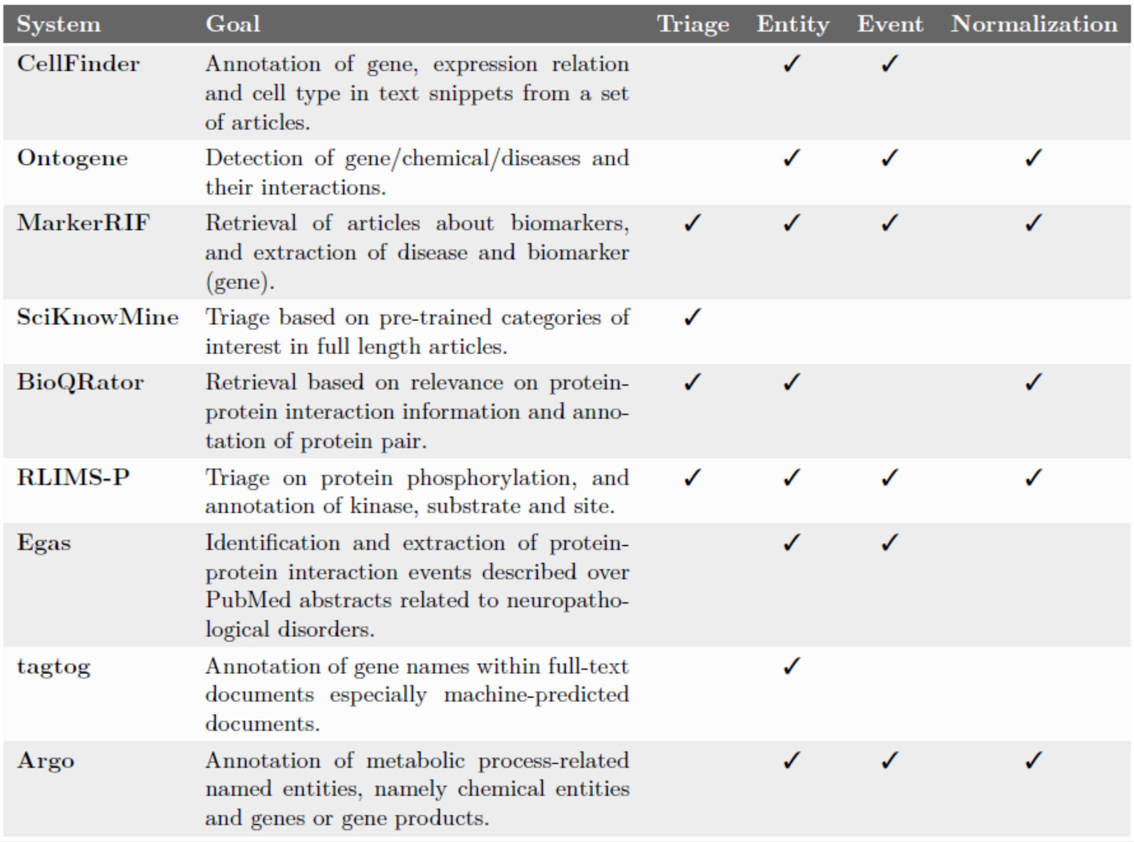
```
    "Neurodegenerative Diseases"[MeSH Terms] OR "Heredodegenerative Disorders,
Nervous System"[MeSH Terms] AND hasabstract[text] AND English[lang].
```

The documents were then ranked according to their relevance for extracting protein-protein interactions, using a SVM classifier [81] trained on the BioCreative III PPI Article Classification Task data [82]. Finally, the top-ranked 100 documents were selected for the task.

Four curators were selected, and each was assigned 50 documents from the corpus to curate. Curators were asked to annotate 25 of their assigned documents using the available PPI annotation service described above, and the remaining 25 documents without using this service, in order to assess its impact on curation effort. In the first case, curators had to revise the automatically generated annotations, correcting any erroneous concept or relation annotations and adding missing ones. In the second case, curators had to annotate all mentions of protein names and all protein interactions described in each document. The tool recorded the time taken by each curator to curate each document, as well as the number of annotated concepts and relations [83].

6.2. Results

Nine systems participated in the BioCreative IV IAT task, following different approaches and targeting heterogeneous domains of application. Figure 50 presents a summary of the systems that participated in the task, presenting the goals and supported tasks. Overall, only 4 systems provide integrated triage features, with 8 systems supporting entity recognition (5 of those with normalization), and 6 enabling relation/event mining. The systems differed significantly in the followed approaches, in terms of design, implementation and usability. For instance, Argo offers a workflow based solution that supports designing custom processing pipelines, tagtog and SciKnowMine support active learning based on triage and concepts provided interactively, and MarkerRIF is provided as a web browser extension to interact directly with Pubmed's web page articles. On the other hand, CellFinder, BioQRator, RLIMS-P and Ontogene follow a more traditional approach, with common input and output of text mining results with highlighting and sorting/scoring capabilities.



System	Goal	Triage	Entity	Event	Normalization
CellFinder	Annotation of gene, expression relation and cell type in text snippets from a set of articles.		✓	✓	
Ontogene	Detection of gene/chemical/diseases and their interactions.		✓	✓	✓
MarkerRIF	Retrieval of articles about biomarkers, and extraction of disease and biomarker (gene).	✓	✓	✓	✓
SciKnowMine	Triage based on pre-trained categories of interest in full length articles.	✓			
BioQRator	Retrieval based on relevance on protein-protein interaction information and annotation of protein pair.	✓	✓		✓
RLIMS-P	Triage on protein phosphorylation, and annotation of kinase, substrate and site.	✓	✓	✓	✓
Egas	Identification and extraction of protein-protein interaction events described over PubMed abstracts related to neuropathological disorders.		✓	✓	
tagtog	Annotation of gene names within full-text documents especially machine-predicted documents.		✓		
Argo	Annotation of metabolic process-related named entities, namely chemical entities and genes or gene products.		✓	✓	✓

Figure 50: Summary of the systems that participated in the BioCreative IV IAT task, presenting the target goals and supported tasks, namely triage, entity recognition, event mining, and entity normalization [59]. Note that when we participate in this task, Egas did not yet support normalization.

To properly evaluate the behavior of the various systems, the BioCreative IV Interactive Annotation Task organization committee built a detailed survey to subjectively rank and compare the different tools[38].

This survey covers various aspects regarding curators' satisfaction, such as:

a) Overall reaction;
b) Comparison with similar systems;
c) Ability to complete tasks;
d) Application design;
e) Learning to use the application;
f) System usability.

The answers to each of the 23 questions were scaled from 1 to 5, where 1 is very bad and 5 very good. In the end, the obtained evaluation results were averaged and grouped into three categories: recommendation, rating and experience. Figure 51 presents the

[38] Available at http://ir.cis.udel.edu/biocreative/survey2.html

satisfaction results obtained by Egas, comparing the achieved results with the remaining participating systems. As we can see, Egas presented very satisfying results in the three categories from the four curators, with, on average, 4.5 points in recommendation and 4.75 points in rating and experience, outperforming other participating solutions in terms of satisfaction.

Regarding curation time, the application of automatic annotation algorithms significantly contributed to reduced curation times. For 3 of the 4 curators, the curation times were reduced by 1.5 to 4 factors.

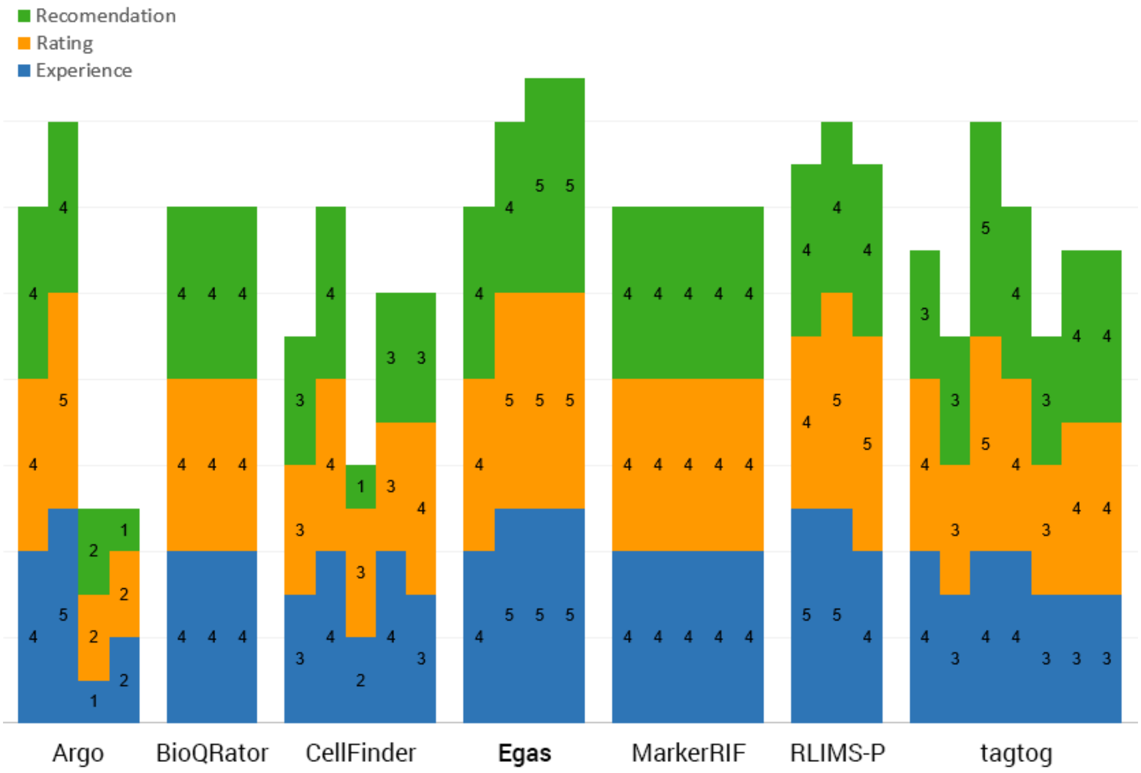


Figure 51: Illustration of the overall satisfaction of curators per system, including user experience (blue), the rating of the system (orange), and the recommendation of the system (green) (adapted from [59]).

However, we also observed that automatic services may contribute to biased annotations, since curators tend to be influenced by automatic results, by accepting or performing slight changes without throughout analysis and reflection. Thus, the annotation guidelines followed on manual and assisted documents may diverge significantly, which must be carefully considered in any annotation task. Moreover, provided annotation guidelines and developed automatic tools should follow the same guidelines, in order to reduce mistakes as much as possible. For instance, if the automatic tool provides species names as part of protein names, and the annotation guidelines indicate otherwise, the final corpus can be easily inconsistent and with serious annotation mistakes, seriously degrading the final IAA (Inter Annotator Agreement).

Another important aspect is the speed of representing documents with respective annotations, since curators frequently change between documents to confirm, verify and/or correct performed annotations. Thus, reduced loading times provide a smooth and sophisticated navigation and interaction with the system. In order to compare our HTML, CSS and Javascript-based approach with similar SVG-based systems, we compared the document loading times of Egas with one of the most used tools, Brat. By selecting one of the largest documents from the CRAFT corpus [37], which contains 3461 concept annotations, we measured the time spent until the document is completely and successfully presented to the user. Thus, brat required, on average, almost 10 seconds to load the document, and Egas only spent, on average, 4 seconds to present the same full-text article. We repeated this experiment with different documents, with different number of concept annotations and sizes in order to obtain more results. Figure 52 presents the performance results of the two systems. Thus, our approach presents an improvement, in some articles, of almost 4 times in terms of document representation.

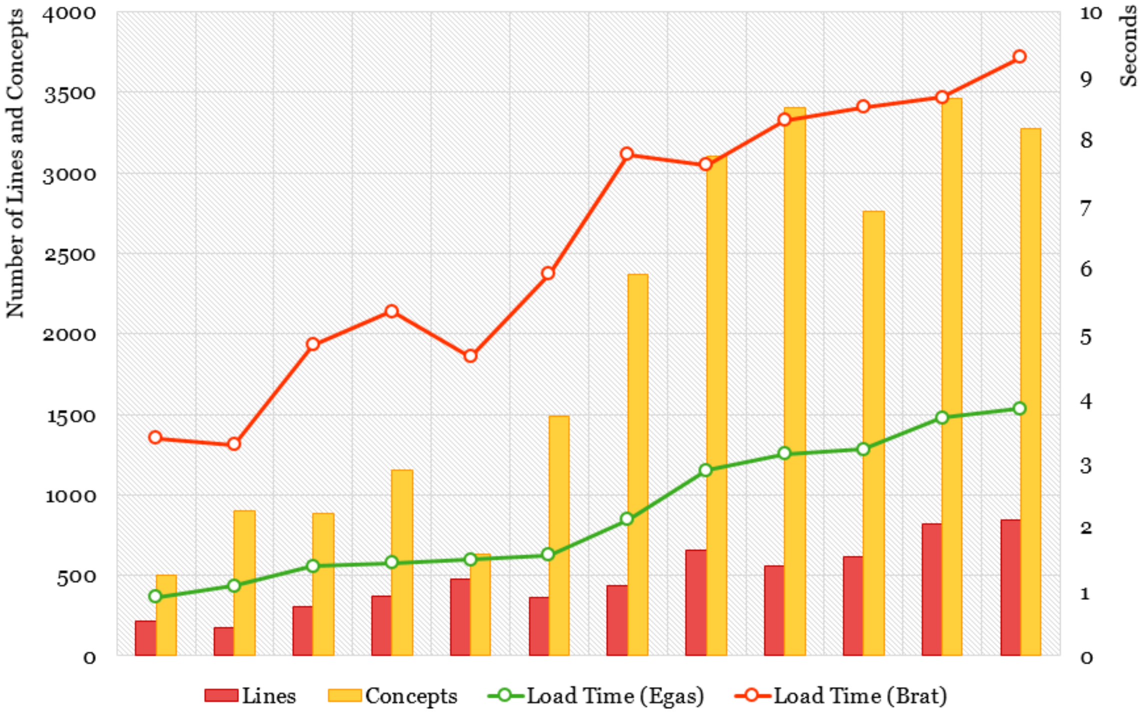


Figure 52: System performance of Egas (green) and Brat (red) when loading different articles (using Mozilla Firefox[39]).

Overall, Egas presented consistent and convincing results in terms of usability, reliability and performance, outperforming the behavior of other solutions. Additionally, Egas showed superior document processing and representation speeds, which is a significant added value and contribution to a smoother annotation process. Moreover, the significantly different times between manual and assisted curation promise to

[39] https://www.mozilla.org/firefox/

facilitate such tasks, allowing faster and better curation of larger amounts of documents, both abstracts and full-texts. Such results together with the previously presented features and characteristics, show that Egas is a state-of-the-art solution to perform a variety of biocuration tasks, ready to grow and to be integrated with any major platform to support information generation and keep current databases properly updated in a consistent way.

Chapter 7

Conclusion

The main goal of this thesis was the study, design and development of a ready-to-use solution that covers the limitations found on the biomedical literature curation domain. Thus, this work started with a careful study and analysis of the existing solutions in order to understand which barriers and limitations we can find in the available systems. After assessing existing system and user requirements we came up with a architecture for the desired solution. As a result, we presented Egas (http://bioinformatics.ua.pt/egas), a web-based platform for biomedical text mining and collaborative curation.

The user interface was carefully developed targeting simplicity and intuitive interactions, through inline document visualization, filtering, insertion and deletion of annotations and relations. Additionally it provides a rich set of features to support the complete workflow of knowledge curation, such as integrated project management, import and export features from local and remote servers, automatic annotation services and innovative real-time collaboration. On the other hand, the tool follows an "annotation-as-a-service" paradigm in which document collections, users, configurations, annotations, data storage, as well as the tools for document processing and text mining, are all centrally managed.

The developed solution is based on the idea of projects, consisting of a curation or document annotation task, performed on a collection of documents, by a team of expert curators, and considering a set of target concept and relation types defined by the curation guidelines and project administrators. Moreover, the system provides easy to use project configurations, allowing concepts and relations to be easily defined and configured according to the desired task. Furthermore, Egas can be further extended and adapted to a given curation task, by exploiting external automatic annotation tools that are available as web-services. Curators can then revise and tune the annotations provided by such tools, allowing them to perform their task more efficiently.

In order to evaluate the applicability of Egas, we participated in the BioCreative IV IAT task, through the identification of biomolecular events described in PubMed abstracts related to neuropathological disorders, including protein-protein interactions, protein expression and post-translational modifications. To support this task, an automatic service for PPI annotation was integrated in Egas, providing annotations for protein concepts, relations between proteins, relations marking equivalent protein mentions, and active words that could potentially indicate the presence of PPIs in a sentence. When evaluated by expert curators, Egas presented convincing results in terms of usability, reliability and performance, outperforming the behavior of other solutions. As a matter of fact, the application of automatic annotation services considerably reduced the curation times of the expert curators.

In conclusion, Egas showed superior document processing and representation speeds, which is a significant added value and contribution to a smoother annotation process. Overall, Egas presents various advantages for the biomedical community, streamlining the collaboration between supervisors and curators, and simplifying the setup and on-demand configuration of the annotation task, using integrated knowledge bases and automatic annotation services. These contributions, together with the presented results, enhance the reliability of Egas in terms of a state-of-the-art solution to perform biomedical text curation providing several features and services that help curators keep the current knowledge bases properly updated.

7.1. Future Work

Though Egas already provides a rich set of features that make it an innovative solution with many advantages for biocuration community, we believe that it can be a baseline for a more advanced platform to support the interactive mining of biomedical information. Thus, there are many features that can be integrated in Egas to further improve it, delivering enhanced assistance to Biocurators, such as:

- Support more knowledge bases for normalization (e.g., ontologies from BioPortal, Uniprot and UMLS);
- Support more input and output formats (e.g., PDF and SQL);
- Support more automatic annotation services (e.g., DDI and events), including confidence values of predicted annotations.

Regarding to concept annotation, there are some features that we intend to support, namely:

- Event extraction through an unique and easier to understand inline representation;
- Provide features to add notes and text passages as supporting information for concept, relation and/or event annotations;
- Add search capabilities to documents;

- Add documents annotations comparison features;
- Develop features to support colorblind curators.

On the other hand, document triage features may be also integrated, through the development of a master index and respective services with automatically annotated concepts, relations and scores of multiple document ranking strategies.

Additionally, to simplify document management, we also intend to support document comparison, and provide features to search for specific terms in the set of documents in the project. Finally, in order to promote wider usage, we intend to create a standalone server and respective configuration scripts for simplified distribution and installation in local machines.

Bibliography

[1] M. Hilbert, P.L., *The world's technological capacity to store, communicate, and compute information*. Science, 2011.

[2] Rowley, J., *The wisdom hierarchy: representations of the DIKW hierarchy*. Journal of Information Science 33, 2007

[3] Robert Blumberg, S.A., *The problem with unstructured data*. DM REVIEW, 2003. **13**: p. 42–49.

[4] Fayyad, U.M., G. Piatetsky-Shapiro, and P. Smyth, *From data mining to knowledge discovery: an overview*, in *Advances in knowledge discovery and data mining*, M.F. Usama, et al., Editors. 1996, American Association for Artificial Intelligence. p. 1-34.

[5] Bateman, A., *Curators of the world unite: the International Society of Biocuration*. Bioinformatics, 2010. **26**(8): p. 991.

[6] Bourne, P.E. and J. McEntyre, *Biocurators: contributors to the world of science*. PLoS Comput Biol, 2006. **2**(10): p. e142.

[7] Burge, S., T.K. Attwood, A. Bateman, T.Z. Berardini, M. Cherry, et al., *Biocurators and biocuration: surveying the 21st century challenges*. Database (Oxford), 2012. **2012**: p. bar059.

[8] Alex, B., C. Grover, B. Haddow, M. Kabadjov, E. Klein, et al., *Assisted curation: does text mining really help?* Pacific Symposium on Biocomputing Pacific Symposium on Biocomputing, 2008: p. 556-67.

[9] Karamanis, N., R. Seal, I. Lewin, P. McQuilton, A. Vlachos, et al., *Natural language processing in aid of FlyBase curators*. BMC bioinformatics, 2008. **9**: p. 193.

[10] Bairoch, A., *The future of annotation/biocuration*. Nature Precedings, 2009(713).

[11] Hirschman, L., G.A. Burns, M. Krallinger, C. Arighi, K.B. Cohen, et al., *Text mining for the biocuration workflow*. Database (Oxford), 2012. **2012**: p. bas020.

[12] Bolchini, D., A. Finkelstein, V. Perrone, and S. Nagl, *Better bioinformatics through usability analysis*. Bioinformatics, 2009. **25**(3): p. 406-412.

[13] Hunter, L. and K.B. Cohen, *Biomedical language processing: what's beyond PubMed?* Mol Cell, 2006. **21**(5): p. 589-94.

[14] Cohen, A.M. and W.R. Hersh, *A survey of current work in biomedical text mining*. Brief Bioinform, 2005. **6**(1): p. 57-71.

[15] Ackoff, R.L., *From data to wisdom*. Journal of applied systems analysis, 1989. **15**: p. 3-9.

[16] Vlachos, A., *Semi-supervised learning for biomedical information extraction*. 2010.

[17] Swanson, D.R., *Medical Literature as a Potential Source of New Knowledge*. Bulletin of the Medical Library Association, 1990. **78**(1): p. 29-37.

[18] Trifiro, G., A. Fourrier-Reglat, M.C. Sturkenboom, C. Diaz Acedo, J. Van Der Lei, et al., *The EU-ADR project: preliminary results and perspective*. Stud Health Technol Inform, 2009. **148**: p. 43-9.

[19] Oliveira, J.L., P. Lopes, T. Nunes, D. Campos, S. Boyer, et al., *The EU-ADR Web Platform: delivering advanced pharmacovigilance tools*. Pharmacoepidemiol Drug Saf, 2013. **22**(5): p. 459-67.

[20] Martinez-Cruz, C., I.J. Blanco, and M.A. Vila, *Ontologies versus relational databases: are they so different? A comparison*. Artificial Intelligence Review, 2012. **38**(4): p. 271-290.

[21] Boutet, E., D. Lieberherr, M. Tognolli, M. Schneider, and A. Bairoch, *UniProtKB/Swiss-Prot*. Methods Mol Biol, 2007. **406**: p. 89-112.

[22] Ashburner, M., C.A. Ball, J.A. Blake, D. Botstein, H. Butler, et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium*. Nat Genet, 2000. **25**(1): p. 25-9.

[23] Lassila, O. and R.R. Swick, *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation, 1999.

[24] Belleau, F., M.A. Nolin, N. Tourigny, P. Rigault, and J. Morissette, *Bio2RDF: towards a mashup to build bioinformatics knowledge systems*. J Biomed Inform, 2008. **41**(5): p. 706-16.

[25] Jupp, S., J. Malone, J. Bolleman, M. Brandizi, M. Davies, et al., *The EBI RDF platform: linked open data for the life sciences*. Bioinformatics, 2014. **30**(9): p. 1338-9.

[26] Schuemie, M.J., M. Weeber, B.J. Schijvenaars, E.M. van Mulligen, C.C. van der Eijk, et al., *Distribution of information in biomedical abstracts and full-text publications*. Bioinformatics, 2004. **20**(16): p. 2597-604.

[27] He, Y. and M. Kayaalp, *A Comparison of 13 Tokenizers on MEDLINE*. Tech. Rep. LHNCBC-TR-2006-003, 2006.

[28] Sætre, R., K. Yoshida, A. Yakushiji, Y. Miyao, Y. Matsubyashi, et al., *AKANE system: protein-protein interaction pairs in BioCreAtIvE2 challenge, PPI-IPS subtask*. Proceedings of the Second BioCreative Challenge Workshop, 2007: p. 209-212.

[29] Tomanek, K., J. Wermter, and U. Rahn, *A Reappraisal of Sentence and Token Splitting for Life Sciences Documents*. Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics, Pts 1 and 2, 2007. **129**: p. 524-528.

[30] Tsuruoka, Y., Y. Tateishi, J.D. Kim, T. Ohta, J. McNaught, et al., *Developing a robust part-of-speech tagger for biomedical text*. Advances in informatics, 2005: p. 382–392.

[31] Liu, H., T. Christiansen, W.A. Baumgartner, Jr., and K. Verspoor, *BioLemmatizer: a lemmatization tool for morphological processing of biomedical text*. J Biomed Semantics, 2012. **3**: p. 3.

[32] Sagae, K., *Dependency parsing and domain adaptation with LR models and parser ensembles*, in *Eleventh Conference on Computational Natural Language Learning*. 2007, Association for Computational Linguistics: Prague, Czech Republic. p. 1044–1050.

[33] Miyao, Y. and J. Tsujii, *Feature forest models for probabilistic HPSG parsing*. Computational Linguistics, 2008. **34**(1): p. 35-80.

[34] Petrov, S. and D. Klein, *Improved Inference for Unlexicalized Parsing*, in *HLT-NAACL*. 2007.

[35] Miyao, Y., K. Sagae, R. Saetre, T. Matsuzaki, and J. Tsujii, *Evaluating contributions of natural language parsers to protein-protein interaction extraction*. Bioinformatics, 2009. **25**(3): p. 394-400.

[36] Qian, L. and G. Zhou, *Tree kernel-based protein-protein interaction extraction from biomedical literature*. J Biomed Inform, 2012. **45**(3): p. 535-43.

[37] Verspoor, K., K.B. Cohen, A. Lanfranchi, C. Warner, H.L. Johnson, et al., *A corpus of full-text journal articles is a robust evaluation tool for revealing*

differences in performance of biomedical natural language processing tools. Bmc Bioinformatics, 2012. **13**.

[38] Ananiadou, S. and J. Mcnaught, *Text Mining for Biology And Biomedicine*. 2005: Artech House.

[39] Zhou, G., J. Zhang, J. Su, D. Shen, and C. Tan, *Recognizing names in biomedical texts: a machine learning approach*. Bioinformatics, 2004. **20**(7): p. 1178-90.

[40] Tsuruoka, Y., J. McNaught, J. Tsujii, and S. Ananiadou, *Learning string similarity measures for gene/protein name dictionary look-up using logistic regression*. Bioinformatics, 2007. **23**(20): p. 2768-2774.

[41] Jimeno-Yepes, A.J. and A.R. Aronson, *Knowledge-based biomedical word sense disambiguation: comparison of approaches*. BMC Bioinformatics, 2010. **11**: p. 569.

[42] Agirre, E. and P.G. Edmonds, *Word sense disambiguation: Algorithms and applications*. 2006.

[43] Navigli, R., *Word Sense Disambiguation: A Survey*. Acm Computing Surveys, 2009. **41**(2).

[44] Weeber, M., J.G. Mork, and A.R. Aronson, *Developing a test collection for biomedical word sense disambiguation*. Journal of the American Medical Informatics Association, 2001: p. 746-750.

[45] Pustejovsky, J., J. Castano, R. Saurí, A. Rumshinsky, J. Zhang, et al., *Medstract: creating large-scale information servers for biomedical libraries*, in *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain*. 2002. p. 85–92.

[46] Raileanu, D., P. Buitelaar, S. Vintar, and J. Bay, *Evaluation Corpora for Sense Disambiguation in the Medical Domain*, in *LREC*. 2002.

[47] Hahn, U., K.B. Cohen, Y. Garten, and N.H. Shah, *Mining the pharmacogenomics literature--a survey of the state of the art*. Brief Bioinform, 2012. **13**(4): p. 460-94.

[48] Swanson, D.R., *Complementary structures in disjoint science literatures*, in *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*. 1991, ACM: Chicago, Illinois, USA. p. 280-289.

[49] Ananiadou, S., S. Pyysalo, J. Tsujii, and D.B. Kell, *Event extraction for systems biology by text mining the literature*. Trends Biotechnol, 2010. **28**(7): p. 381-90.

[50] Cohen, A.M. and W.R. Hersh, *A survey of current work in biomedical text mining*. Briefings in Bioinformatics, 2005. **6**(1): p. 57-71.

[51] Simpson, M. and D. Demner-Fushman, *Biomedical Text Mining: A Survey of Recent Progress*, in *Mining Text Data*, C.C. Aggarwal and C. Zhai, Editors. 2012, Springer US. p. 465-517.

[52] Zhu, F., P. Patumcharoenpol, C. Zhang, Y. Yang, J. Chan, et al., *Biomedical text mining and its applications in cancer research*. Journal of Biomedical Informatics, 2013. **46**(2): p. 200-211.

[53] Cohen, K.B. and L. Hunter, *A critical review of PASBio's argument structures for biomedical verbs*. BMC Bioinformatics, 2006. **7 Suppl 3**: p. S5.

[54] Kim, J.-D., T. Ohta, S. Pyysalo, Y. Kano, and J.i. Tsujii, *Overview of BioNLP'09 shared task on event extraction*, in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. 2009, Association for Computational Linguistics: Boulder, Colorado. p. 1-9.

[55] Kim, J.-D., S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, et al., *Overview of BioNLP Shared Task 2011*, in *Proceedings of the BioNLP Shared Task 2011 Workshop*. 2011, Association for Computational Linguistics: Portland, Oregon. p. 1-6.

[56] Nédellec, C., R. Bossy, J.D. Kim, J.J. Kim, T. Ohta, et al. *Overview of BioNLP Shared Task 2013*

Proceedings of the BioNLP Shared Task 2013 Workshop. Association for Computational Linguistics.

[57] Krallinger, M., F. Leitner, C. Rodriguez-Penagos, and A. Valencia, *Overview of the protein-protein interaction annotation extraction task of BioCreative II*. Genome Biol, 2008. **9 Suppl 2**: p. S4.

[58] Arighi, C.N., P.M. Roberts, S. Agarwal, S. Bhattacharya, G. Cesareni, et al., *BioCreative III interactive task: an overview*. BMC Bioinformatics, 2011. **12 Suppl 8**: p. S4.

[59] Matis-Mitchell, S., P. Roberts, C.O. Tudor, and C.N. Arighi, *BioCreative IV Interactive Task*, in *Fourth BioCreative Challenge Evaluation Workshop*. 2013: Bethesda, MD, USA. p. 190-203.

[60] Stenetorp, P., S. Pyysalo, G. Topi, #263, T. Ohta, et al., *BRAT: a web-based tool for NLP-assisted text annotation*, in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 2012, Association for Computational Linguistics: Avignon, France. p. 102-107.

[61] Salgado, D., M. Krallinger, M. Depaule, E. Drula, A.V. Tendulkar, et al., *MyMiner: a web application for computer-assisted biocuration and text annotation*. Bioinformatics, 2012. **28**(17): p. 2285-7.

[62] Rak, R., A. Rowley, W. Black, and S. Ananiadou, *Argo: an integrative, interactive, text mining-based workbench supporting curation*. Database (Oxford), 2012. **2012**: p. bas010.

[63] Wei, C.H., H.Y. Kao, and Z. Lu, *PubTator: a web-based text mining tool for assisting biocuration*. Nucleic Acids Res, 2013. **41**(Web Server issue): p. W518-22.

[64] Rinaldi, F., S. Clematide, G. Schneider, M. Romacker, and T. Vachon, *ODIN: An Advanced Interface for the Curation of Biomedical Literature*. 2010.

[65] Comeau, D.C., R. Islamaj Dogan, P. Ciccarese, K.B. Cohen, M. Krallinger, et al., *BioC: a minimalist approach to interoperability for biomedical text processing*. Database (Oxford), 2013. **2013**: p. bat064.

[66] Maglott, D., J. Ostell, K.D. Pruitt, and T. Tatusova, *Entrez Gene: gene-centered information at NCBI*. Nucleic acids research, 2011. **39**(Database issue): p. D52-7.

[67] Federhen, S., *The NCBI Taxonomy database*. Nucleic acids research, 2012. **40**(Database issue): p. D136-43.

[68] Hewett, M., D.E. Oliver, D.L. Rubin, K.L. Easton, J.M. Stuart, et al., *PharmGKB: the Pharmacogenetics Knowledge Base*. Nucleic acids research, 2002. **30**(1): p. 163-5.

[69] Manning, C.D., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. 2008: Cambridge University Press. 496.

[70] Osmani, A., *Developing Backbone.js Applications*. 2013: O'Reilly Media.

[71] Smith, B., M. Ashburner, C. Rosse, J. Bard, W. Bug, et al., *The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration*. Nat Biotechnol, 2007. **25**(11): p. 1251-5.

[72] Sioutos, N., S. de Coronado, M.W. Haber, F.W. Hartel, W.L. Shaiu, et al., *NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information*. J Biomed Inform, 2007. **40**(1): p. 30-43.

[73] Federhen, S., *The NCBI Taxonomy database*. Nucleic Acids Res, 2012. **40**(Database issue): p. D136-43.

[74] Natale, D.A., C.N. Arighi, W.C. Barker, J.A. Blake, C.J. Bult, et al., *The Protein Ontology: a structured representation of protein forms and complexes*. Nucleic Acids Res, 2011. **39**(Database issue): p. D539-45.

[75] Degtyarenko, K., P. de Matos, M. Ennis, J. Hastings, M. Zbinden, et al., *ChEBI: a database and ontology for chemical entities of biological interest*. Nucleic Acids Res, 2008. **36**(Database issue): p. D344-50.

[76] Schriml, L.M., C. Arze, S. Nadendla, Y.W. Chang, M. Mazaitis, et al., *Disease Ontology: a backbone for disease semantic integration*. Nucleic Acids Res, 2012. **40**(Database issue): p. D940-6.

[77] Sayers, E. and V. Miller, *Overview of the E-utility Web Service (SOAP)*, in *Entrez Programming Utilities Help*. 2010 [Updated 2012 May 31], Bethesda (MD): National Center for Biotechnology Information (US).

[78] Nunes, T., D. Campos, S. Matos, and J.L. Oliveira, *BeCAS: biomedical concept recognition services and visualization*. Bioinformatics, 2013. **29**(15): p. 1915-6.

[79] Campos, D., S. Matos, and J.L. Oliveira, *A modular framework for biomedical concept recognition*. BMC Bioinformatics, 2013. **14**: p. 281.

[80] Campos, D., S. Matos, and J.L. Oliveira, *Gimli: open source and high-performance biomedical name recognition*. BMC Bioinformatics, 2013. **14**: p. 54.

[81] Matos, S. and J.L. Oliveira, *Classification methods for finding articles describing protein-protein interactions in PubMed*. J Integr Bioinform, 2011. **8**(3): p. 178.

[82] Krallinger, M., M. Vazquez, F. Leitner, D. Salgado, A. Chatr-Aryamontri, et al., *The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text*. BMC Bioinformatics, 2011. **12 Suppl 8**: p. S3.

[83] Campos, D., J. Lourenço, T. Nunes, R. Vitorino, and P. Domingues. *Egas-Collaborative Biomedical Annotation as a Service*, in Fourth BioCreative Challenge Evaluation Workshop, Bethesda, Maryland, USA, Oct. 2013, p. 254–259;