Computer Methods and Programs in Biomedicine 159 (2018) 125–134

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Development of an information retrieval tool for biomedical patents

Tiago Alves[a,b], Rúben Rodrigues[a], Hugo Costa[b], Miguel Rocha[a,*]

[a] Centre Biological Engineering, University of Minho, Braga 4710-057, Portugal
[b] Silicolife Lda, Braga 4715-387, Portugal

ARTICLE INFO

Article history:
Received 21 August 2017
Revised 7 February 2018
Accepted 12 March 2018

Keywords:
Biomedical text mining
Information retrieval
Information extraction
Patents
PDF to text conversion

ABSTRACT

Background and objective: The volume of biomedical literature has been increasing in the last years. Patent documents have also followed this trend, being important sources of biomedical knowledge, technical details and curated data, which are put together along the granting process. The field of Biomedical text mining (BioTM) has been creating solutions for the problems posed by the unstructured nature of natural language, which makes the search of information a challenging task. Several BioTM techniques can be applied to patents. From those, Information Retrieval (IR) includes processes where relevant data are obtained from collections of documents. In this work, the main goal was to build a patent pipeline addressing IR tasks over patent repositories to make these documents amenable to BioTM tasks.

Methods: The pipeline was developed within @Note2, an open-source computational framework for BioTM, adding a number of modules to the core libraries, including patent metadata and full text retrieval, PDF to text conversion and optical character recognition. Also, user interfaces were developed for the main operations materialized in a new @Note2 plug-in.

Results: The integration of these tools in @Note2 opens opportunities to run BioTM tools over patent texts, including tasks from Information Extraction, such as Named Entity Recognition or Relation Extraction. We demonstrated the pipeline's main functions with a case study, using an available benchmark dataset from BioCreative challenges. Also, we show the use of the plug-in with a user query related to the production of vanillin.

Conclusions: This work makes available all the relevant content from patents to the scientific community, decreasing drastically the time required for this task, and provides graphical interfaces to ease the use of these tools.

1. Introduction

Biomedical Text Mining (BioTM) is a sub-field of Text Mining that has been applied to biological and biomedical literature, seeking to automate the processing of the huge amounts of data that are generated every day including papers, reports and patents [1,2], among others. Since these texts are written in natural language, in an unstructured form (without annotations about the text structure and available entities), searching for structured information by hand is quite time-consuming [3]. To automate this process, BioTM approaches become crucial, allowing the extraction of meaningful knowledge from texts, as a way to formulate scientific hypotheses more easily [4,5]. Towards that aim, BioTM makes use of different knowledge fields, such as Statistics, Artificial Intelligence or Management sciences, combined with numerous text analytics components as Information Retrieval (IR), Information Extraction (IE), Natural Language Processing (NLP), among others (Fig. 1) [6,7].

IR methods deal with information resources, gathering metadata and documents from an extensive collection of sources. On the other hand, IE addresses the extraction of pertinent structured information from human readable documents, using computer tools [8].

NLP includes a set of computational and linguistic methods to be applied over texts, as spelling correction or document indexing. These systems can split texts in tokens or words (in a process called tokenization), perform word grammatical classification (using part-of-speech (POS)), group words by families (with lemmatization processes), find the syntactic structure of expressions, among others [9].

As it is shown in Table 1, several BioTM platforms has been developed by the scientific community using distinct programming languages, and applying different IR and IE methodologies (the table only reviews a few to illustrate the diversity of the field).

* Corresponding author.
E-mail address: mrocha@di.uminho.pt (M. Rocha).

https://doi.org/10.1016/j.cmpb.2018.03.012
0169-2607/

Table 1
Some BioTM platforms, their programming languages and a brief review on BioTM functionalities.

BioTM platform	Programming Language	BioTM functionalities
@Note [10]	Java	Information retrieval and extraction of biomedical information from literature, detecting entities and relations between them in raw text. Desktop application only. Covers IR and IE.
iHOP [11]	PHP, AJAX, and Javascript	Information retrieval from MEDLINE abstracts related to protein interactions and genes information. Focuses mainly on building networks of biological knowledge.
Neji [12]	Java	Identification of bio-entities in free text (IE). Focuses on NER.
ABNER [13]	Java	Tagging genes, proteins and other bio-entities in free text (information extraction). Only performs NER.
GATE [14]	Java	General-purpose TM framework, used to develop some BioTM applications. Identification of chemical and drug bio-entities as well as chemical formulas in free text (IE).
UIMA [15]	Java, C++	General-purpose TM framework, used to develop some BioTM applications.
MyMiner [16]	PHP, AJAX, and Javascript	Web application for bio-entity tagging (IE). Does not cover other IE or IR tools.

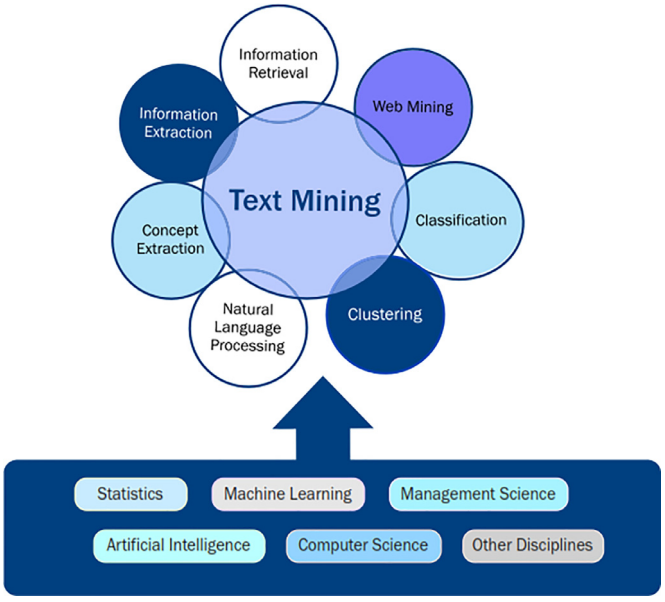

Fig. 1. Text mining, the text analytical components contributing to this effort and the knowledge fields from complementary areas. Adapted from [7].

@Note2,[1] developed by the University of Minho and the SilicoLife company is among these efforts, being an extended and reformulated version of the original @Note application, originally released in 2009 [10]. @Note2 is a purely Java BioTM framework, which uses a relational database and is based on a plug-in architecture, allowing the development of new tools/methodologies in the BioTM field. Structurally, this software can be the basis for the development of new services, algorithms or graphical components, following the clear separation between core BioTM libraries and the GUI tools, the user interface section of @Note2 (Fig. 2).

The core libraries are organized in three main functional modules: the Publication Manager Module (PMM), which searches for documents on online repositories (IR Search process) and downloads their respective full-text documents (IR Crawling process); the Corpora Module (CM), which performs corpora management, creates and applies IE processes over these corpora, such as Named Entity Recognition (NER) and Relation Extraction (RE), and provides a manual curation system; and the Resources Module (RM), which allows creating and editing lexical resources to be used in IE processes.

The user interface tools allow a friendlier and more productive interaction with the user, that is able to configure and use each functionality [10]. Therefore, developers can create new algorithms and tools, which can be used by users with minimum effort. At the same time, the core libraries implementing the main methods can be used in the development of other applications besides @Note, providing an API for third-party developers.

So, in recent years, alongside with the interest given to the information provided in scientific papers, there has been a growing interest in patents. To explore their data is crucial to understand many biological fields, since they have detailed technical descriptions of inventions put together as part of the granting process. Furthermore, that information is rarely disclosed in other published documents [17–19].

Patents are classified into hierarchical categories which can be useful when the interest is only in a specific knowledge field (such as biomedicine) [20]. Due to the huge number of patents generated every year, there are several databases able to store these data. For instance, the World Intellectual Property Organization (WIPO) database had 2.7 million patents registered only in 2014 [3,21,22].

Since every patent has a geographical region where it is active, there are both world-wide and localized patents. In both cases, there are specific patent databases. The PATENTSCOPE from WIPO or esp@cenet from European Patent Office (EPO) are included in the former group, with patents with granted protection to all the countries in the world. The j-PlatPat from Japan Patent Office (JPO) or PatFT from the United States Patent and Trademark Office (USPTO) are databases with patents from Japan and United States, respectively, included in the latter group [18].

Although several patent databases are available to search and retrieve documents, the access to large amounts of data is restricted. Several studies were made, recently, trying to address this problem, using information from several patent sections.

Heifets and Jurisica built SCRIPDB (a chemical structure database) used USPTO bulk download from Google servers to retrieve patents available since 2010 [23]. Papadatos et al. used patent data from USPTO, EPO and WIPO, together with titles and abstracts from the JPO (processed in the XML format) to build SureChEMBL, a database with compound's structures. For patent data processing, they used IFI Claims, a third-party global patent database aggregating patent information from over 40 national and international data sources into a single repository [24].

A related effort on patent IR is the TREC Chemical IR Track (TREC-CHEM), an initiative that has started in 2009 with the support of NIST (National Institute for Standards and Technology) [25]. This effort focuses on the evaluation of IR methods and knowledge discovery of stored information on chemical patents and articles. Although being an interesting approach to benchmark IR systems that seek documents based on relevance for given queries, it does not address the problems of patent recovery, mainly related to the full text documents, tasks that will be handled by our proposed software tools.

A possible solution for full text retrieval is using Portable Document Format (PDF) files of each published patent, which can be accessed on patent databases. However, these files have their data saved into scanned pages (usually image files in BMP, TIFF,

[1] http://anote-project.org/.

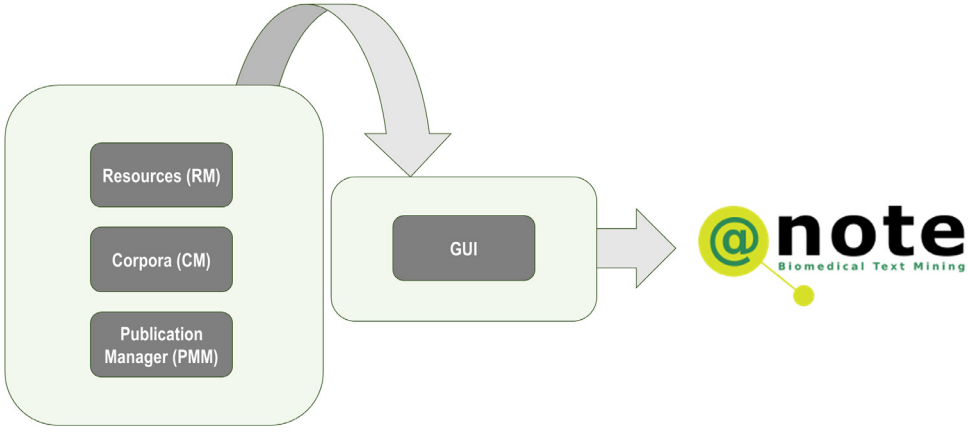

Fig. 2. @Note2 main structure. The @Note2 GUI has methods that implement graphically the functions from the core text mining libs running in the back-end.

PNG or GIF formats). So, when there is the need to extract information from this type of files (such as chemical structures), it is necessary to convert these images into structured representations of standard chemical files format (such as Simplified Molecular Input Line Entry System (SMILES) strings) [26].

To read scanned text, technologies such as Optical Character Recognition (OCR) can be used [27], allowing the retrieval of machine-coded, readable, editable and searchable data from these files. OCR is a computer science technology that associates the image of a character to its symbolic identity. OCR uses an offline approach to character recognition, in which the recognition is done after the completion of writing or printing and not done when characters are being drawn [28]. Since OCR systems are based on patterns (numbers, letters or punctuation characters), the first step is to "teach" the machine about these patterns. For that purpose, there are training sets with characters grouped into classes used to analyze and compare with new examples, assigning them to the best-scored class [29].

This technology can be summarized in two main processes: the character extraction and the character recognition process. In the first step, the previous learned patterns are applied to delimit words or individual symbols and then, in the second step, to identify each symbol [30]. This is a complex process with several robust algorithms and techniques to analyze distortions, style variations, translation and rotation movements, background noise, etc [29].

Given the scenario depicted in the previous paragraphs, we realize that a few software tools exist allowing users to search and recover patents from repositories and handle their content. However, to the best of the authors' knowledge, no existing free software tool currently is able to perform searches for patents over different repositories, integrating the results of those queries, and obtaining the full text files for the resulting patents. Also, the connection of patent retrieval with downstream BioTM analysis of the content of full texts is almost nonexistent.

In this work, the objective was to develop and test a patent pipeline, added as a new plug-in to @Note2, able to retrieve patents' metadata and full texts, using OCR when needed to extract text from the PDF files. This will make patent data amenable to be searched and used as an information source for IE processes already available in @Note2. We will provide two cases to validate the methods in our pipeline and demonstrate the usefulness of the developed plug-in.

2. Methods

The patent pipeline can be organized into four different tasks. It can search for patent identifiers and retrieve patent metadata (as title and authors), download the published patent PDF file, and, finally, apply PDF to text conversion methodologies to those files. Each task was structured into a module with specific inputs and outputs. Thus, sources to search and retrieve patent IDs, to search for metadata about each patent, and to return the patent file(s) in PDF format were configured as components of the *search sources module, metainformation sources module* and *retrieval sources module*, respectively. The PDF to text conversion methodologies used were organized in the *PDF conversion module* (Fig. 3).

To make the patent pipeline fully available, some specific access keys, which can be obtained from the registration services of the different repositories must be used. Although not all components require those keys, each component has restrictions when multiple requests are made. So, to get large amounts of information, all components must be configured. In this way, when the limits imposed by a specific component are reached, the others can still be used. This approach is followed until all data are retrieved or all components for each module are used.

To start the search process, input keyword(s) are required, which may be, for instance, biomedical entities as chemicals, genes or diseases. These keywords are then processed by the *search sources module*. In this module, two popular search engines (the *Custom Search* API from Google and the *Bing Search* API from Microsoft) were used. These APIs return patent identifiers from Google Patents, a database with around 87 million of patents from 17 countries [31].

Alongside with those two search engines, a local patent repository, the *Power User Gateway (PUG)* from PubChem, and the *Open Patent Services (OPS) web services* API from EPO were also used. The patent repository was built using the bulk data files from USPTO since January 2005, having a schedule for weekly updates, getting all the new available US patents. These patents were parsed and saved into specific data structures that can be easily accessed. Since all the patents are properly indexed, and the system was built using a RESTful architecture, all the US patent identifiers related with the given keywords can be retrieved without restrictions (Fig. 4).

When the input is a chemical name, a PubChem Compound Identifier (CID), an InCHI key or a SMILE string, the PUG system from PubChem is autonomously used and, through the web interface, all the patents documented as being related with that chemical in PubChem can be retrieved. For this last module, an access token is required.

The *metainformation sources module* receives a set of patent identifiers and input, and returns the invention title, authors, publication date, an external link to a patent database entry (if available) and the abstract to each patent. When available,

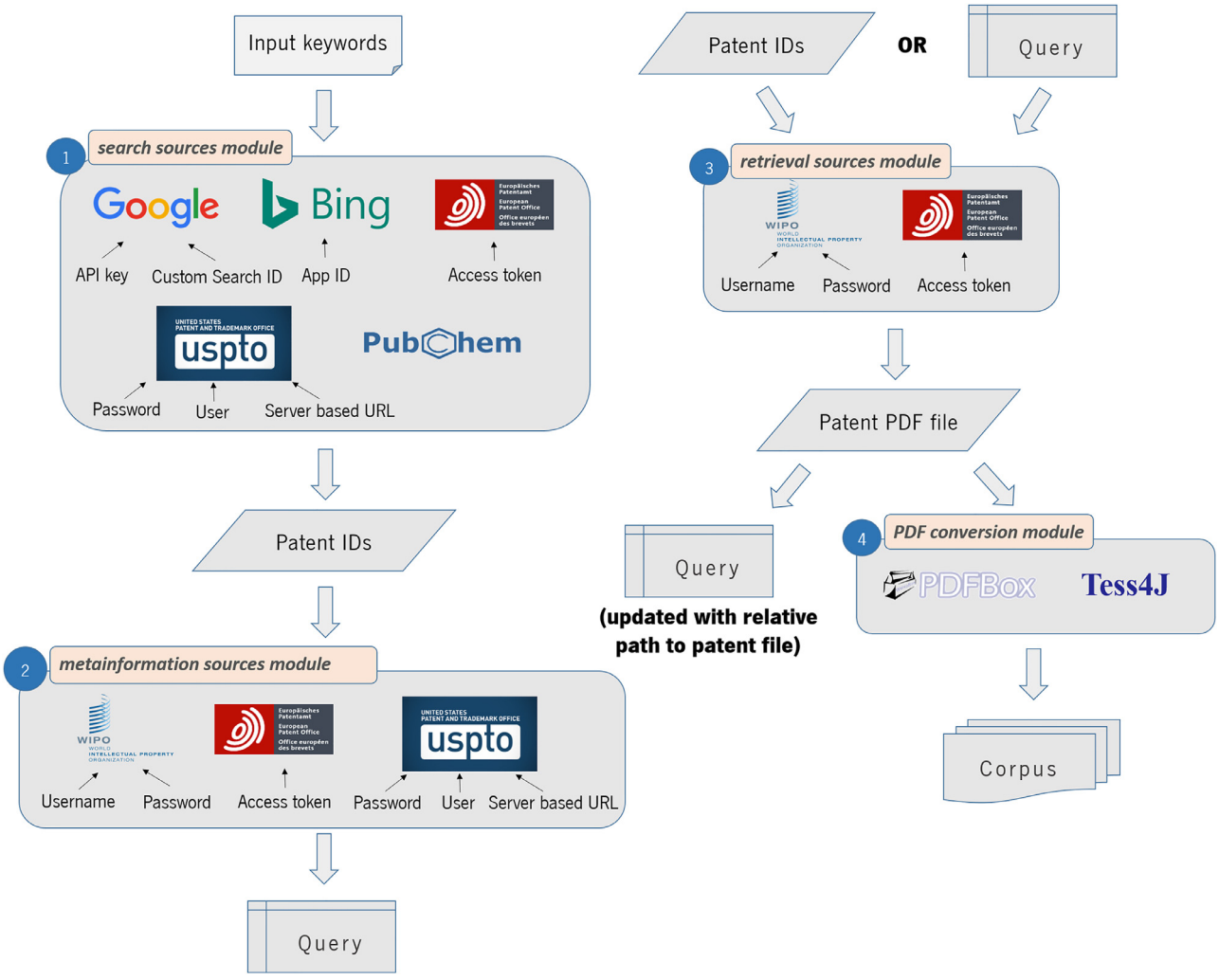

Fig. 3. Summary of the designed patent pipeline, where numbers represent the process flow.

Fig. 4. Example of the used URI to access the local repository with patent from USPTO since 2005, searching for patents using specific keywords.

the description and claims sections are also extracted. To avoid repetitions, the patent family is extracted previously and only one identifier is used to retrieve metadata, being the others saved as external references for that patent.

The metadata obtained as a result of this module is stored into a *query*, a data structure from @Note2 to save this type of information (Fig. 5). Three different services can be used in this module: the *PATENTSCOPE web service* API from WIPO, the *OPS web service* API from EPO and our local USPTO database, all capable of returning patents' metadata, given a set of identifiers.

The *retrieval sources module* takes as input a set of identifiers or a query data structure, and returns the patent's PDF files, saving the path for each into the *query* (Fig. 5). This module includes two APIs from the *metainformation sources module* using different configurations: the *PATENTSCOPE web service* API from WIPO and the *OPS web service* API from EPO.

Then, lastly, the *PDF conversion module* takes all the files returned from the previous one, and extracts their text. As shown in Fig. 5, this allows the creation of a *corpus* data structure, allowing to run IE methods able to retrieve meaningful information from these texts, for instance, performing NER or RE.

In this module, alongside with the version 2.0.1 of the *Apache PDFBox* library (implemented on @Note2), the *Tess4J*, version 3.2.1 (developed by Quan Nguyen) was configured implementing *Tesseract* (an OCR algorithm from Google), and also a hybrid method combining these two methodologies. The *Apache PDFBox* allows to extract the Unicode text available on PDF documents. The hybrid method allows a previous PDF treatment, improving their quality to be processed by the *Tess4J* system.

Since our patent repository has all the full-text content from each of US patents since 2005, for those, the text can automati-

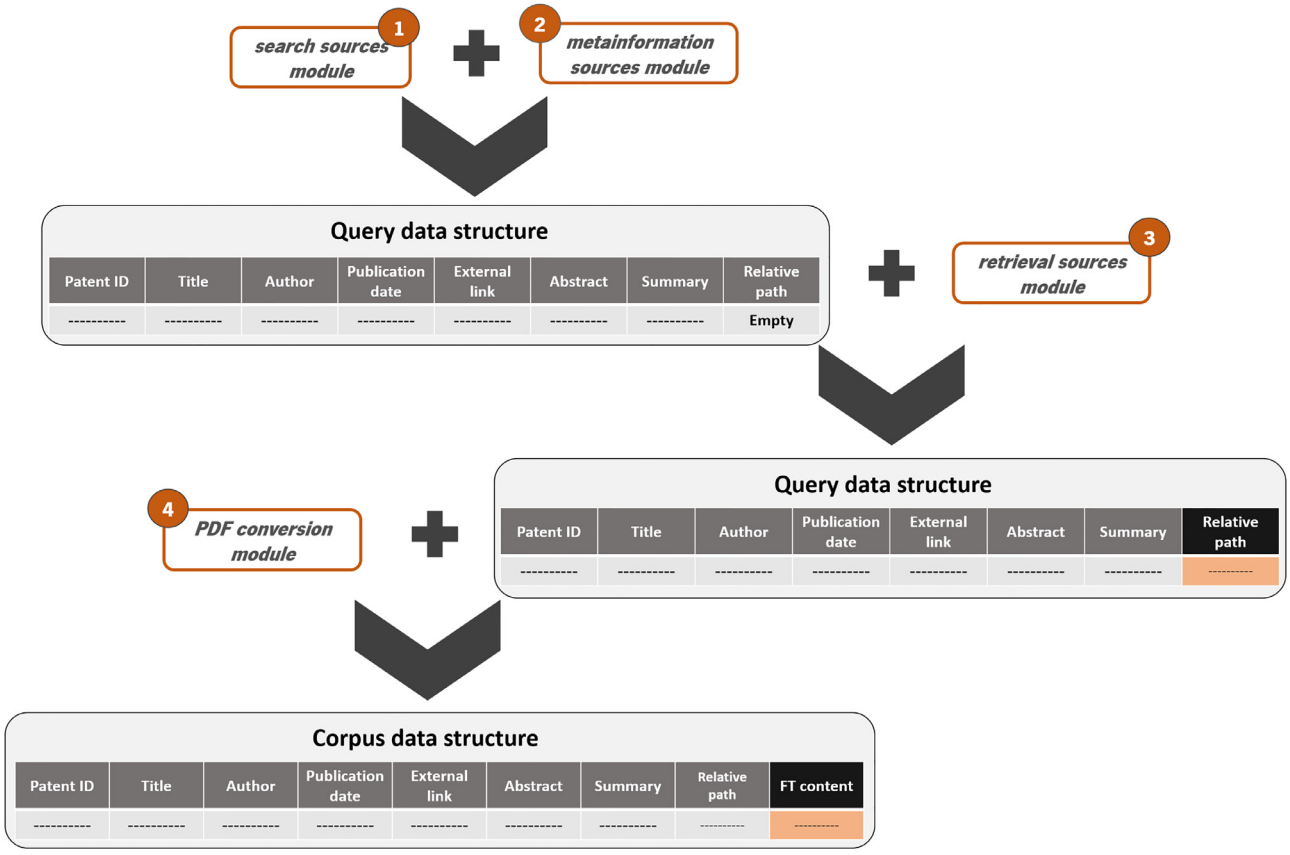

Fig. 5. Creation and update process for *query* and *corpus* data structures. The numbers represent the modules of the pipeline and their flow. The orange *query* data field represents the update process of the original *query*, while the orange *corpus* data field represents the field that turns the *corpus* into a different data structure.

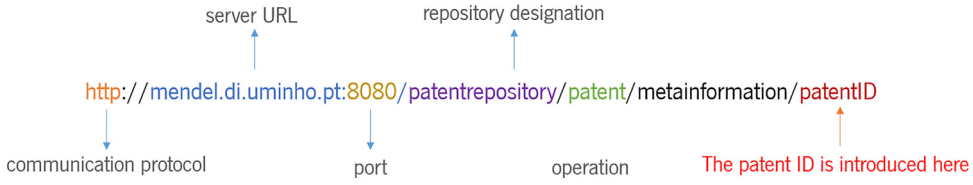

Fig. 6. Example of the used URI to access the local repository with patent from USPTO since 2005, getting metainformation for specific patent IDs.

cally be extracted from our repository, saving processing time and using all the available sources (Fig. 6).

On @Note2, the patent handling features were inserted in different core libraries. The patent ID search and metadata retrieval were added as a new IR Search process called "*Patent Search*", while the patent PDF file download was added as a new IR Crawling process and the new PDF to text conversion methods were put into the Corpora Module as a pre-processing to corpora creation (Fig. 7).

3. Results

The pipeline is materialized by an @Note2 plug-in allowing the search for patents from Google Patents, USPTO and esp@cenet repositories. This plug-in has two different steps and so, two different panes were built (Fig. 8). The first pane has two different fields where the keywords and the *query* name can be inserted. Usually, it is common to leave the query filled with the predefined text (**[KEYWORDS]:[ORGANISM]:[Date]**), since it is a regular expression that produces a specific name to that query taking into account the search date.

The second pane is the configurations one, where all the different components can be configured to run our pipeline. When we try to save any of the access keys, a verification is made, assuring that all the components are correctly configured. Here, since there is also a similar pane in the @Note2 Preferences section, the table can be pre-filled with all the previously configured components and saved to the database. In those cases, the configurations can also be edited.

Two different case studies will be described, the first more focused on allowing to validate the patent IR pipeline and tools, while the second will be addressed to show the usefulness of the @Note plugin and its interface, together with the potential of its integration with other @Note tools.

In the first case study, the complete training set for the BioCreative V CHEMDNER task will be used. In the second, we will use a query relative to the patents registered related to vanillin production. Each case study will be described in one of the next sections.

3.1. BioCreative V CHEMDNER task

The BioCreative CHEMDNER tasks were created to foster developments on the BioTM field. They are based on the imple-

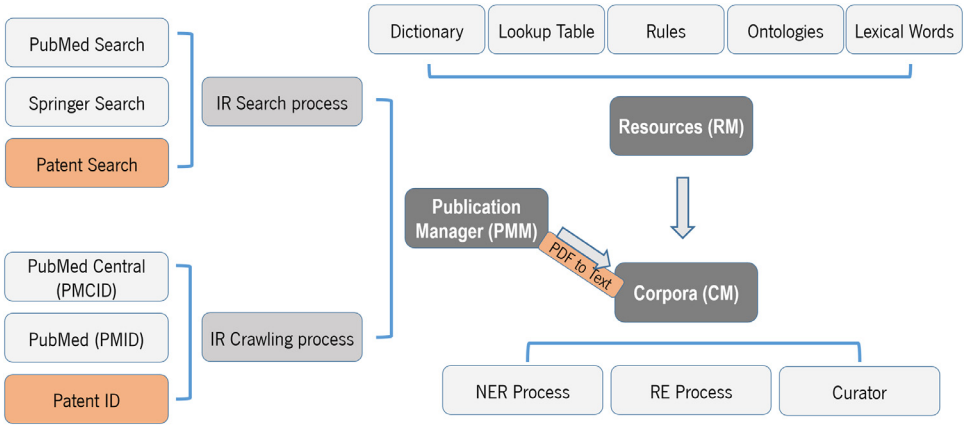

Fig. 7. @Note2 structure with patent pipeline implementations. The orange boxes represent the new components added.

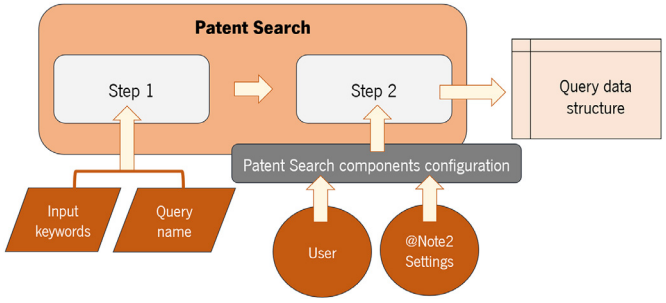

Fig. 8. @Note2 Patent Search plug-in. The pipeline uses input keywords, the *query* name and configurations provided by the user or by @Note2 settings to search for patent IDs and to download patent metadata.

mentation of IE processes to several chemical publications, finding chemical mentions on papers and, more recently, on patent documents. For that, sets with previously annotated data are provided to be the gold standard corpora to train and test systems created by the participating teams.

So, since these challenges contain a large known number of patents, we used one of them to test the patent pipeline, namely considering the data from the training set of the BioCreative V CHEMDNER task. The dataset comprises 7000 patents, their identifiers, respective titles and abstract sections. Thus, using that set, bypassing the search sources module, an exact number of documents are expected.

As shown in Fig. 9, the pipeline was applied taking each of the 7000 identifiers in the dataset. Skipping the first module, the remaining will be executed, taking these identifiers as input, to obtain both the metadata and the full texts of the respective patents. Then, the abstract section from the dataset can be also used to validate our pipeline.

To that, each patent abstract coming directly from the BioCreative V CHEMDNER dataset, and the respective patent text obtained from the patent pipeline as described above, were properly tokenized and compared. In this comparison, we used the Smith-Waterman algorithm, a local Dynamic Programming algorithm, to evaluate the matches. Based on the number of tokens that match exactly on the two texts, we can calculate performance metrics as precision, recall and F1 score. Through those metrics, it is possible to infer the amount of conversion errors, as well as to verify the amount of documents correctly downloaded.

Through the metainformation sources module, metadata was extracted for 6979 patents, which represents a success rate of 99.7%. From those, we get 6440 abstracts, showing that patent abstracts can also be obtained from patent databases, being a quick source of information to several BioTM approaches, similar to what is already done with other scientific publications.

Since our interest is on patent full texts, the patent pipeline follows to the next component. From the 7000 patents, 6967 patent PDFs were downloaded, representing a success rate of 99.5%. After retrieving the patents, the next step was to verify which patents are written in English to exclude old patents (that were then reissued) or those written in a non-English language (being the original PDF files saved on the database, without any translation). Through this process, 2456 patents were excluded from our analysis, remaining 4511.

Then, the Dynamic Programming algorithm was applied to those patents. The evaluation metrics were calculated and the precision values, in means, show that 83.9% of the converted text relative to the abstract is effectively the same for both texts. This is important because it is possible to infer that there is a low percentage of unrecognized characters or even breaks in the recognized words. This conclusion was also supported by the recall values that were, in means, 69.8% and by F1 score, which is 76.6% for all the analyzed documents. Both values showed also a small variance with a low standard deviation (around 10%) (see Fig. 10).

Of course, that value would be changed if we accepted all the PDF to text conversion files. In that case, we obtain, a mean of 89.8% of precision, 39.5% of recall and 54.8% of F1 score. This overall evaluation shows that our system is not able to verify the content of database files, searching in the databases for English-only patents. So, when a patent file is available on database, it is downloaded even if it is written in another language, influencing our results.

To test the informative capacity of patent full-texts, a simple IE task of Named Entity Recognition was performed using a dictionary of chemicals, the JoChem [32]. This dictionary is composed by 278,578 chemical terms and respective synonyms. These terms refer to small molecules and drugs from several sources. To improve the dictionary, rule-based term filtering, manual check of highly frequent terms, and disambiguation rules were applied.

So, applying the NER process to abstracts, we obtained 6493 annotations (which represents a mean of 2 chemical entities for each patent), while using full texts we get 1,401,142 annotations (which represents a mean of 414 entities for each patent). These values represent a huge difference between the information available on the abstracts and full-texts, providing a raw measure of the interest in obtaining full texts.

To process all these files, the whole pipeline took around 3 days using a PC with an i7 960 @ 3.2 GHz processor and 16GB of RAM.

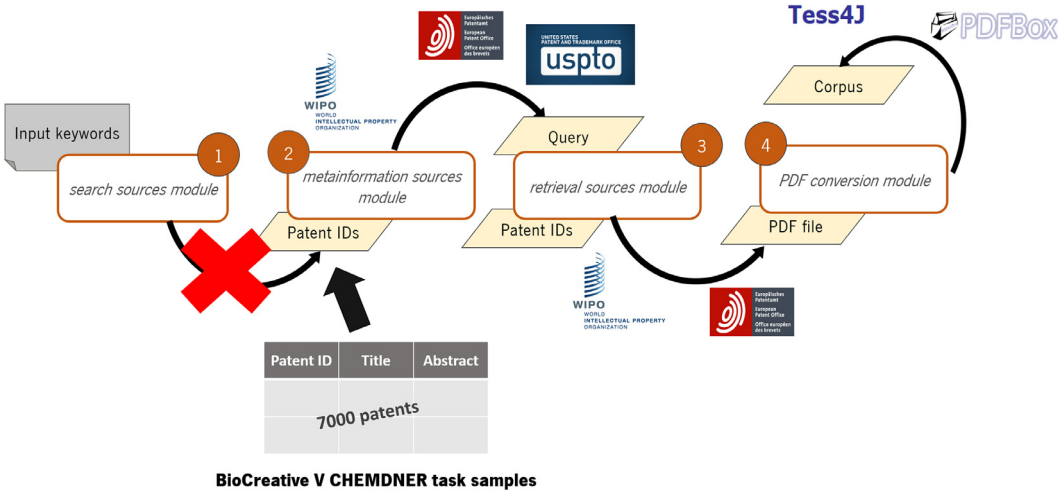

Fig. 9. Injection of the BioCreative V CHEMDNER task training set data into patent pipeline, replacing the *search sources module*.

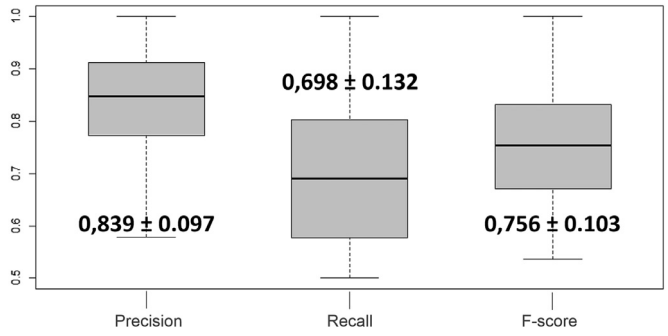

Fig. 10. Boxplots for the evaluation metrics of the PDF to text conversion process. The mean and standard deviation are given in bold.

3.2. Vanillin production case study

To verify our pipeline running a real case study, we decided to make a quick search on patents related with vanillin production. The process was run on the developed @Note2 plug-in, being the main steps illustrated in Fig. 11. As we may find on PubChem, vanillin (PubChem CID: 1183) is the primary component of the extract of the vanilla bean. It can be naturally or industrially produced, to be used as a flavoring agent.

So, as shown in Fig. 11, we used the built patent pipeline on @Note2 to search for “vanillin”. From this search. we expected to obtain all the results related with this compound (directly or indirectly), without any restriction. As a result of that search, we found 7215 patents. All those patents were correctly filled with metadata (in English), and, from those, 7087 patent were filled also with the respective abstracts.

Since the real number of patents related with vanillin is not known, a quick search on PubChem was made, returning 18,863 patents. This large difference in the number of patents can be explained, since these data comprises all the different patent identifiers associated with vanillin. However, a single patent can be registered on different countries with distinct identifiers. In the developed patent pipeline, patents from the same family are grouped into a single patent to avoid repetitions (as can be seen on the results from Fig. 12), and, thus, our results avoid redundancy.

Then, using all the abstracts from vanillin related patents, the same NER task that we run for the BioCreative V CHEMDNER case study was also applied (again using the JoChem dictionary). From that run, we annotated 38,954 entities, which represents a mean of 6 annotations for each patent.

If we intended to obtain all the data from these patents, the next steps from the patent pipeline could be used, and several BioTM approaches would be available to be applied.

4. Discussion

Recently, patents have been a target for BioTM techniques since they are a great source of information for many fields. Based on @Note2, IR search and crawling processes were designed and implemented, allowing the search and retrieval of patent information from several databases, extending the domain of scientific/ technical texts handled by this framework. The retrieved information includes the patent metadata and the respective full text documents, in PDF format. Also, new improvements were made to the @Note2 PDF to text conversion system, allowing a better conversion of patent files to get patent full-text data in a machine-readable format.

Testing the patent IR processes proposed in this work with the training set from the BioCreative V CHEMDNER task shows that 99,7% were filled with patent metadata. There are some patents that are stored in databases in their native language and, for these ones, the full-text content is not meaningful. Using the new PDF to text system on the English documents, we got around 81% of F1-score, which represents a good capacity of our system to transform this type of files.

Using these texts as a study corpus, a simple IE task of named-entity recognition for chemical compounds was made, using a dictionary-based approach. We annotated both the abstract section and the full text content. The results showed that the full text-content is, approximately, 200 times richer in annotations than the abstract section.

To adapt our pipeline into a real case study, a quick search of vanillin production related patents was also made. We got 7215 documents, all filled with metadata and 98,2% with the abstract section included. This case was also used to showcase the potential of the user interface provided by the developed plugin for @Note2.

The main innovation of this work was the creation of new IR processes applied to patents, surpassing common problems related to searching and retrieving those documents, allowing also the posterior implementation of several IE techniques to those texts. Indeed, most of the BioTM systems (reviewed in the Introduction) do not support processing of patents, and open-source software systems for patent retrieval are mostly non-existent.

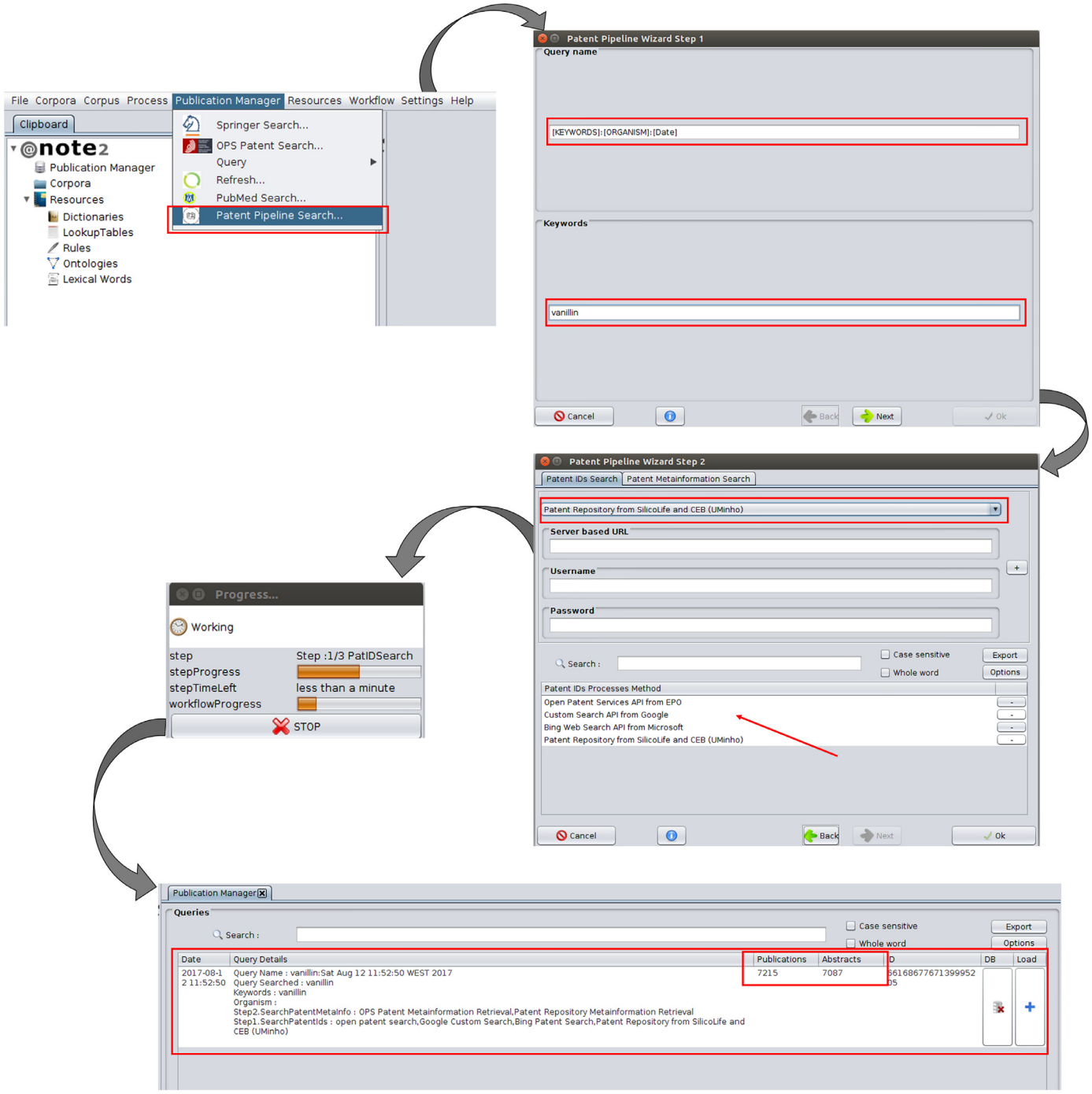

Fig. 11. Patent pipeline running a search for vanillin. In @Note2, the pipeline is opened clicking on Patent Pipeline Search button (Pane in the upper left corner).

Since @Note2 is an open-source software, this framework opens doors to the community to take advantage of all sections of the published patents with biological relevance more easily, and without the need to expend large amounts of time browsing several databases. Regarding @Note2, the integration of these tools allows developing an extensive set of text mining pipelines over patents, which were only possible for scientific articles so far.

In the development of this work, we took into account some of the most important features for patent retrieval systems, namely to incorporate different sorts of patent search processes, to use knowledge of the domain, to use multiple sources of data, and to provide visualisation of the search results [33].

The work is still limited for the restricted access of many search engines and databases, which require authentication and impose limits on the amounts of information it is possible to collect. These issues are important limitations to be addressed in future work. Another aspect of possible improvement for this work is the evaluation of the system. Indeed, the metrics of recall and precision have been disputed recently, and some authors propose the notion of risk [34], an aspect to consider to improve the evalution of the proposed system.

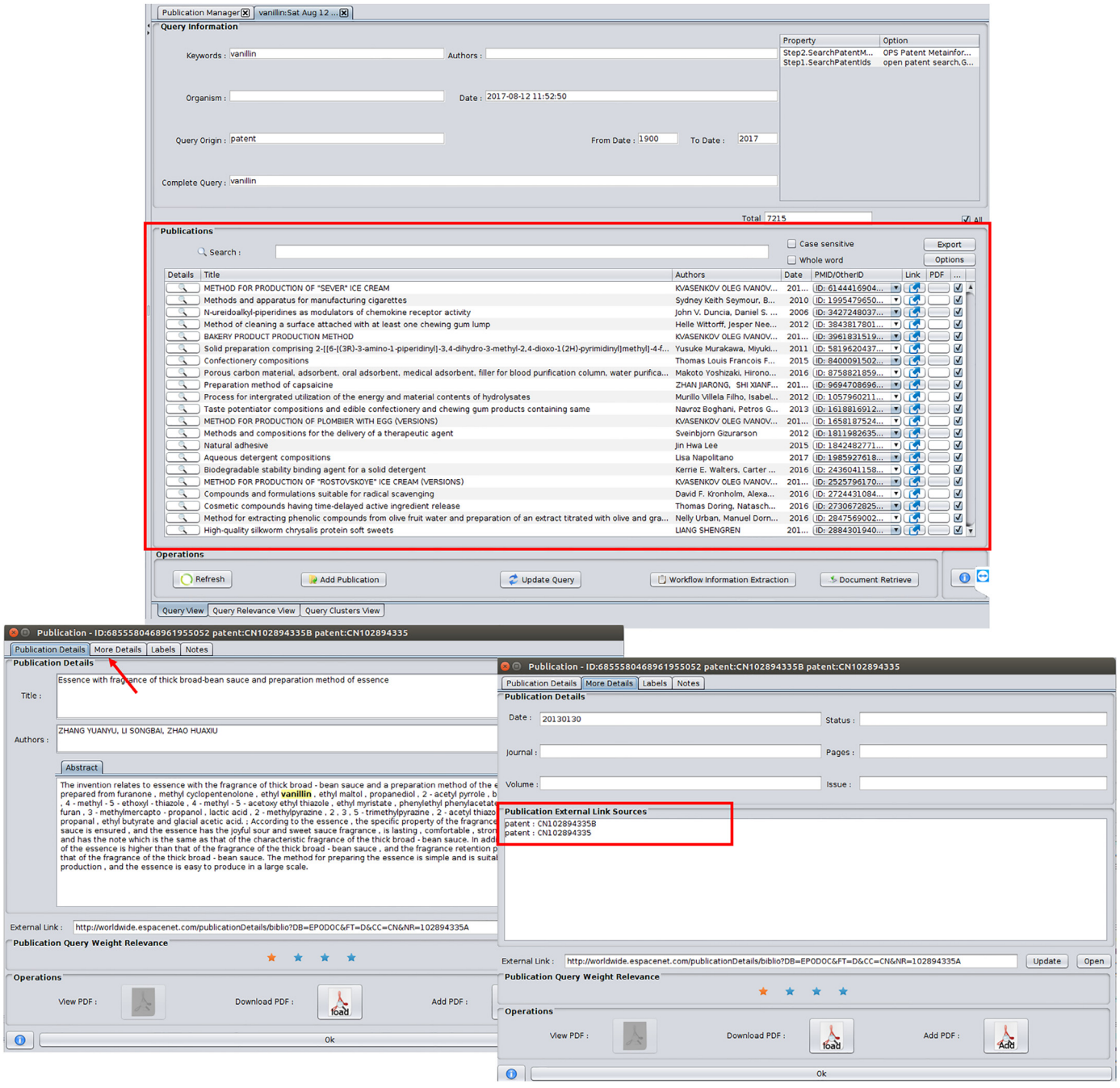

Fig. 12. Patent pipeline running a search for vanillin. From the results, a single patent can be selected to show some data as the Publication ***External Link Sources*** on the ***More Details tab***.

Acknowledgments

This work is co-funded by the Programa Operacional Regional do Norte, under the "Portugal2020", through the European Regional Development Fund (ERDF), within project SISBI- Ref[a] NORTE-01-0247-FEDER-003381.

This study was also supported by the Portuguese Foundation for Science and Technology (FCT) under the scope of the strategic funding of UID/BIO/04469/2013 unit and COMPETE 2020 (POCI-01-0145-FEDER-006684) and BioTecNorte operation (NORTE-01-0145-FEDER-000004) funded by European Regional Development Fund under the scope of Norte2020 - Programa Operacional Regional do Norte.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2018.03.012.

References

[1] A. Faro, D. Giordano, C. Spampinato, Combining literature text mining with microarray data: advances for system biology modeling, Brief. Bioinform. 13 (1) (2012) 61–82, doi:10.1093/bib/bbr018. http://bib.oxfordjournals.org/content/13/1/61.full.pdf.

[2] R. Klinger, C. Kolarik, J. Fluck, M. Hofmann-Apitius, C.M. Friedrich, Detection of IUPAC and IUPAC-like chemical names, Bioinformatics 24 (13) (2008) i268–i276, doi:10.1093/bioinformatics/btn181. http://bioinformatics.oxfordjournals.org/content/24/13/i268.full.pdf.

[3] C. Wu, J.M. Schwartz, G. Brabant, S.L. Peng, G. Nenadic, Constructing a molecular interaction network for thyroid cancer via large-scale text mining of gene and pathway events, BMC Syst. Biol. 9 (Suppl 6) (2015) S5, doi:10.1186/1752-0509-9-S6-S5.
[4] N.R. Smalheiser, D.R. Swanson, Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses, Comput. Methods Programs Biomed. 57 (3) (1998) 149–153. http://www.ncbi.nlm.nih.gov/pubmed/9822851.
[5] N.R. Smalheiser, V.I. Torvik, W. Zhou, ARROSMITH two-node search interface: a tutorial on finding meaningful links between two disparate sets of articles in MEDLINE, Comput. Methods Programs Biomed. 94 (2) (2009) 190–197, doi:10.1016/j.cmpb.2008.12.006. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2693227/pdf/nihms105821.pdf.
[6] K.B. Cohen, L. Hunter, Getting started in text mining, PLoS Comput. Biol. 4 (1) (2008) e20, doi:10.1371/journal.pcbi.0040020.
[7] G. Miner, J. Elder, T. Hill, R. Nisbet, D. Delen, A. Fast, Practical text mining and statistical analysis for non-structured text data applications, Academic Press, 2012.
[8] M. Krallinger, A. Valencia, Text-mining and information-retrieval services for molecular biology, Genome Biol. 6 (7) (2005) 224, doi:10.1186/gb-2005-6-7-224.
[9] A. Clark, C. Fox, S. Lappin, The Handbook of Computational Linguistics and Natural Language Processing., Blackwell Handbooks in Linguistics, Wiley-Blackwell, 2010.
[10] A. Lourenço, R. Carreira, S. Carneiro, P. Maia, D. Glez-Peña, F. Fdez-Riverola, E.C. Ferreira, I. Rocha, M. Rocha, @Note: a workbench for biomedical text mining, J. Biomed. Inform. 42 (4) (2009) 710–720, doi:10.1016/j.jbi.2009.04.002.
[11] R. Hoffmann, A. Valencia, A gene network for navigating the literature, Nat. Genet. 36 (7) (2004) 664, doi:10.1038/ng0704-664.
[12] D. Campos, S. Matos, J.L. Oliveira, A modular framework for biomedical concept recognition, BMC Bioinform. 14 (2013) 281, doi:10.1186/1471-2105-14-281.
[13] B. Settles, ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text, Bioinformatics 21 (14) (2005) 3191–3192, doi:10.1093/bioinformatics/bti475. http://bioinformatics.oxfordjournals.org/content/21/14/3191.full.pdf
[14] H. Cunningham, V. Tablan, A. Roberts, K. Bontcheva, Getting more out of biomedical documents with gate's full lifecycle open source text analytics, PLoS Comput. Biol. 9 (2) (2013) e1002854, doi:10.1371/journal.pcbi.1002854.
[15] D. Ferrucci, A. Lally, UIMA: an architectural approach to unstructured information processing in the corporate research environment, Nat. Lang. Eng. 10 (3–4) (2004) 327–348.
[16] D. Salgado, M. Krallinger, M. Depaule, E. Drula, A.V. Tendulkar, F. Leitner, A. Valencia, C. Marcelle, MyMiner: a web application for computer-assisted biocuration and text annotation, Bioinformatics 28 (17) (2012) 2285–2287, doi:10.1093/bioinformatics/bts435. http://bioinformatics.oxfordjournals.org/content/28/17/2285.full.pdf.
[17] WIPO, Guidelines for Preparing Patent Landscape Reports, 2015.
[18] M.T. Latimer, Patenting inventions arising from biological research, Genome Biol. 6 (1) (2005) 203, doi:10.1186/gb-2004-6-1-203.
[19] WIPO, WIPO Guide to Using Patent Information, 2015.
[20] D. Eisinger, G. Tsatsaronis, M. Bundschus, U. Wieneke, M. Schroeder, Automated patent categorization and guided patent search using IPC as inspired by MeSH and PubMed, J. Biomed. Semant. 4 (Suppl 1) (2013) S3, doi:10.1186/2041-1480-4-S1-S3.
[21] Z. Lu, PubMed and beyond: a survey of web tools for searching biomedical literature, Database 2011 (2011) baq036, doi:10.1093/database/baq036.
[22] WIPO, World Intellectual Property Indicators, 2015, World Intellectual Property Organization - Economics and Statistics Division, 2015. http://www.wipo.int/edocs/pubdocs/en/wipo_pub_941_2015.pdf.
[23] A. Heifets, I. Jurisica, SCRIPDB: a portal for easy access to syntheses, chemicals and reactions in patents, Nucleic Acids Res. 40 (Database issue) (2012) D428–D433, doi:10.1093/nar/gkr919.
[24] G. Papadatos, M. Davies, N. Dedman, J. Chambers, A. Gaulton, J. Siddle, R. Koks, S.A. Irvine, J. Pettersson, N. Goncharoff, A. Hersey, J.P. Overington, SureChEMBL: a large-scale, chemically annotated patent document database, Nucleic Acids Res. 44 (D1) (2016) D1220–D1228, doi:10.1093/nar/gkv1253.
[25] M. Lupu, F. Piroi, J. Huang, J. Zhu, J. Tait, Overview of the TREC 2009 chemical IR track, in: Proceedings of the 18th Text Retrieval Conference, 2009.
[26] J. Park, G.R. Rosania, K.A. Shedden, M. Nguyen, N. Lyu, K. Saitou, Automated extraction of chemical structure information from digital raster images, Chem. Cent. J. 3 (2009) 4, doi:10.1186/1752-153X-3-4.
[27] A.M. Asif, S.A. Hannan, Y. Perwej, M.A. Vithalrao, An overview and applications of optical character recognition, Int. J. Adv. Res.Sci. Eng. 3 (7) (2014).
[28] D. Álvarez, R. Fernández, L. Sánchez, Stroke-based intelligent character recognition using a deterministic finite automaton, Log. J. IGPL 23 (3) (2015) 463–471.
[29] L. Eikvil. Optical character recognition. 1993. *citeseer. ist. psu. edu/142042*.
[30] R. Holley, How good can it get? Analysing and improving OCRaccuracy in large scale historic newspaper digitisation programs, D-Lib Mag. 15 (2009).
[31] Google, About Google patents, 2017. https://support.google.com/faqs/answer/6390996.
[32] K.M. Hettne, R.H. Stierum, M.J. Schuemie, P.J. Hendriksen, B.J. Schijvenaars, E.M. Mulligen, J. Kleinjans, J.A. Kors, A dictionary to identify small molecules and drugs in free text, Bioinformatics 25 (22) (2009) 2983–2991, doi:10.1093/bioinformatics/btp535. http://bioinformatics.oxfordjournals.org/content/25/22/5202983.full.pdf.
[33] B. Diallo, M. Lupu, Future patent search, in: Current Challenges in Patent Information Retrieval, Springer, 2017, pp. 433–455.
[34] A. Trippe, I. Ruthven, Evaluating real patent retrieval effectiveness, in: Current Challenges in Patent Information Retrieval, Springer, 2017, pp. 143–162.