Integration of software tools in patent analysis

Piotr Masiakowski*, Sunny Wang

Global Patent Department, Sanofi US, 55 Corporate Dr, Bridgewater, NJ 08807, USA

Keywords:
Patent information
Text mining
Data visualization
Software integration

1. Introduction

Patent information is critically important for many aspects of a successful business, but its complexity and sheer amounts pose formidable challenges for its efficient and timely utilization. United States Patent and Trademark Office alone has granted over 8 million patents and Espacenet, the online service at the European Patent Office, offers free access to more than 70 million patent documents worldwide. A review of the Statistics on Patents data available on the World Intellectual Property Organization website [1] reveals a constant growth trend in the numbers of patent publications. The data is spread across many different databases, in a wide variety of formats. In addition, the language of patent documents is often difficult to read and understand.

Many different software tools are available to facilitate and streamline patent analysis [2–9]. In particular, text mining and visualization software offers great promise for revolutionizing the field (for a review, see Ref. [4] and references therein). While no current technology can replace an experienced human analyst, natural language processing (NLP)-based text mining approaches can be used to quickly extract interesting and relevant knowledge from unstructured text, and present it for a final review by the analyst. Such information can be then summarized for the decision makers using statistical analysis and data visualization tools. It would be convenient if the whole process — from the initial searches to the distribution of the final reports — could be performed in a single homogeneous software environment. In practice, however, the best commercial software is often written with generic functionality in mind, so that it can be applied to a wide range of tasks. For example, the Linguamatics I2E text mining software [10] is being used to extract information from such diverse sources as scientific articles and patents on one hand, and news or Twitter feeds on the other. While I2E’s agile NLP-base querying can be easily applied to any text, optimal processing and visualization of such different types of results may require different, specialized software programs. Assuring that the output and input formats for different programs are compatible becomes a major concern and can become a bottleneck in our work.

The purpose of this paper is to describe the use of a custom-configured software integration tool to combine the best features of multiple software systems, producing results far exceeding the capabilities of any one tool alone. We demonstrate the usefulness of software tool integration in a patent landscaping project on antibody–drug conjugate (ADC) therapeutics. ADCs represent a promising technology for targeting powerful toxins to cancer cells to destroy them without harming healthy tissue [11,12]. This is achieved by attaching cell-killing chemicals to monoclonal antibodies which selectively recognize antigens found on the surface of malignant cells. There are thousands of patent publications (e.g. Refs. [13–19]) claiming various combinations of the basic aspects of this technology: antibodies recognizing different proteins, different toxins, different chemical linkers used to combine them, and different types of targeted malignancies. We use three different software systems sequentially — PatBase [20], I2E and Intellixir [21]. We use PatBase patent database to rapidly select a set of relevant patent families using simple Boolean queries. We use the Linguamatics I2E text mining software to automatically categorize this collection of patent documents according to claimed technologies, and the Intellixir System statistical analysis and visualization software to display the results. Custom-developed plug-ins for the PatMOb software [22]; see the Annex for description] are used to transfer the original text corpus from PatBase to I2E, to review and edit I2E results, and to format them for an upload to Intellixir.

* Corresponding author.
E-mail address: piotr.masiakowski@sanofi.com (P. Masiakowski).
2. Selection and formatting of the text corpus

Most patent offices offer free access to the full text of their patent publications, but in practice it is more convenient to use services of one of the patent database vendors who combine documents from many national offices, index and assemble them into patent families, and provide various search tools. PatBase from Minesoft Ltd and RWS Group contains 40 million patent families from over 95 issuing authorities. PatBase offers a user-friendly web interface allowing querying, storage, and web publishing of collections of patent families. Selected patent publications can be also downloaded in a variety of formats for further processing in a downstream application. We use broad queries, designed to maximize recall at the expense of precision, to generate a set of patent families for further analysis. This corpus is then transferred to the I2E text mining software.

I2E can be configured to accept any XML document as a source for indexing. However, the PatBase XML cannot be easily used for that purpose. For example, the entire Claims section of a patent is included in a CDATA (character data) section which is not parsed by a standard XML parser. Individual claims in PatBase XML are separated by various arrangements of HTML tags; pre-processing of this XML with a specially written PatMOb plug-in replaces such arbitrary separators with standard XML tags, allowing recognition of individual claims as separate regions in I2E queries. In addition, PatMOb pre-processing allows further customization of the corpus. For example, in this project we preserve the PatBase patent family structure during the conversion, but keep only the English text of title, abstract and claims of the American and European publications and PCT applications — if available. Documents from other authorities can be repetitive or contain no English text, and are used only for patent families with no US, EP or WO members.

PatBase offers a well designed API for online programmatic access to the basic functionality of the database. PatMOb “PatBase API Access” plug-in connects to the PatBase API, downloads the export XML, and converts it initially to native PatMOb objects (Fig. 1). The PatMOb objects are then written to a file in the I2E XML format, which can be uploaded to the I2E server and used to create a searchable index using an I2E administrator account.

Figure 1 shows the PatBase patent families imported in the standard editor window of PatMOb by the plug-in. The functionality of this window is available to the user through the menu bar, buttons, and the contextual pop-up menu. The pop-up menu is opened by right-click on a tree node; it allows accessing the patent publication at the patent office or an external database in the default browser, removing the node, adding a child node, or changing its highlight color. All this functionality is also available to the plug-ins. For example, the “PatBase API Access” plug-in displays all the patent publications in a patent family, but highlights the ones that were written to I2E XML. When the content of the editor window is saved to the internal database, only the tree structure in the left panel is preserved. The bibliographic data in the INFO tab on the right is fetched in real time from the European Patent Office’s Open Patent Services (OPS) when the user clicks on a patent number. Using the full publication cycle feature assures that when the user clicks on an application that has been granted, the information about the grant is also displayed. We will use the editor window extensively in the next section to edit I2E results.

3. Categorization of patent publications by text mining

Linguamatics describes I2E as “agile, high performance enterprise text mining software”, and indeed the agility and flexibility of its NLP-based querying is remarkable. Its uses in text analysis are practically unlimited; in our project we take advantage of ontologies to categorize patent documents. I2E ontologies are hierarchies defined by classes and macros. Classes can represent any concept, such as proteins, diseases or relationships, and are built into I2E indexes during their creation. Macros have similar functionality, but can be quickly developed by the user and used with any index.

Figure 2 shows the “I2E Pro” interface of I2E. In this interface, complex queries can be developed by arranging graphical elements in the Query Editor. For example, the box with the quotation mark icon represents a phrase containing two items, (i) an alternative of the word “anti” or substring “anti-”, and (ii) the “Human Genes” class. The properties of the phrase are set so that the two items have to be ordered, with no other words between them. The “Human
Genes” class will match any gene synonym included in the NCBI Entrez Gene, or UniProt databases. Since specific antibodies are conventionally designated with prefix “anti-” preceding the gene name, such as “anti-HER2 antibody”, this simple query will identify in our index genes targeted by antibodies.

The default display of the I2E results is an HTML table (Fig. 3), with matches to query items shown in columns, and assertions — combinations of hits in individual documents — in rows. Each row can include a cell showing the color coded keywords in a brief context. Clicking this cell displays the full text of the document with highlighted keywords, in a separate browser window. The display format can be extensively customized at various levels. In addition, the results can be exported in generic formats like XML and CSV, and as files for specific applications — Excel, FreeMind and Cytoscape. These options allow the analyst a rapid overview and optimization of the search and archiving of the results, but are usually not optimal for the final visualization of the conclusions. Also, we find the I2E built-in results curation tool not practical for rapid and efficient editing of large result sets.

PatMoB “I2E Tools” plug-in, developed to facilitate editing of I2E results, parses the I2E XML export file and prompts user for selection of the columns to be used for categorization of patent publications. It automatically annotates patent documents with the corresponding values from the selected columns, and displays them in a standard PatMoB Tree Branch Editor. The resulting list of patent families includes only documents with I2E hits (Fig. 4A). Each patent node in this tree has now children, in this case human genes found in its text. The plug-in extracts from the I2E results file address of the I2E cache, so clicking on the annotation opens the document with I2E-highlighted hits in a default browser window, allowing rapid evaluation of the relevancy of the annotation. Any node of the tree — an annotation, a document, or
a patent family – can be easily removed using the contextual pop-up menu. Editing results in the context of patent families has many advantages. Deletion of a family node allows quick removal of many similar patent equivalents of an irrelevant hit. In addition, the user can decide to delete applications for which patents have been granted, or to keep just one representative publication per family.

The ease and quickness of this process allows it to be repeated multiple times with different variants of the query. Note, for example, that our simple query in Fig. 2, while exhibiting great precision, will fail to find expressions like “an antibody binding to HER2”. A query with an un-ordered phrase containing “antibody” within a few word distance of the “Human Genes” class will work.
but it poses another problem. Many synonyms for human genes are 2- or 3-letter acronyms, so it will retrieve a large number of false positives consisting of short words and irrelevant abbreviations occurring in close proximity to the word “antibody”. In addition, such query will still fail with the following hypothetical, but fairly typical example of patent claims language: “1. An antibody—drug conjugate, wherein the antibody is directed against a cell-surface receptor... 2. The conjugate of claim 1, wherein said receptor is selected from HER2 or KDR.”

I2E supports development of extremely sophisticated queries, and in fact complex multi-queries have been created to parse claim chains [David Milward, Personal communication]. However, the ability to quickly edit the results obtained with multiple “noisy” queries may often be more time-efficient than an effort to optimize the query, and in some cases the analyst’s input cannot be avoided. Note in Fig. 3, for example, that based on Entrez Gene ontology, I2E recognizes the term “CS1” as a synonym for five different genes, and an in-depth investigation is needed to resolve this ambiguity. By enabling such strategies, the use of “I2E Tools” plug-in expands the agility and flexibility of I2E.

Once the initial review and editing of I2E results is finished, Tree Branch Editor’s menu commands can be used to further reorganize the annotated patent family list. Figure 4A–C shows the tree structures from subsequent editor windows during the transition. In the first step, patents are organized by their annotations, rather than patent families. In the tree fragment shown in Fig. 4B, patent publications are categorized by genes targeted by the antibodies they claim. Since the “Human Genes” I2E class used in this example utilizes the PANTHER Classification System [23,24] of gene families and subfamilies, the PANTHER system can be used to organize the genes—and their corresponding patents—into families and subfamilies. Selecting another command from the Editor’s menu will prompt the user to upload a text file defining the annotations hierarchy, and use it to generate the multi-level tree structure shown in Fig. 4C.

Finally, the entire tree can be inserted in an appropriate node of the PatMOB database. In Fig. 5 the Human Genes branch is inserted into the ADC node. The ADC node holds now a set of patent publications organized by the Human Genes ontology, and also three other aspects of the technology: ADC Linkers, Antineoplastic Agents, and Neoplasms, processed in a similar fashion.

4. Presentation of the results

Intellixir System is a hosted web application for statistical analysis of patent and non-patent literature. Many dynamic and interactive displays are available for data visualization. Intellixir offers multiple user levels, so that a “power user” can perform a complex data analysis and provide a client with only the basic access to the interactive display of the results. The Boolean queries in Intellixir are not as powerful as those in I2E, but combining these two applications allows us to take advantage of I2E sophistication in text analysis and Intellixir user-friendly data visualization.

Fig. 6. Intellixir system thesaurus display, showing the ADC tree.
Intellixir offers various custom “filters” for parsing data exported from other applications. It also supports its own native XML format which can be used to import and export data. In addition, sets of Boolean queries can be uploaded to Intellixir as text files in a tab-separated values (TSV) format, to define custom fields and thesauri used to group documents. Such XML and TSV files can be easily generated by a PatMOb plug-in to upload to Intellixir highly customized content.

![Custom fields value co-occurrences analysis in Intellixir.](image1)

![The values of custom field “ADC linkers” for selected affiliations, as Intellixir bubble graph.](image2)
5. Discussion and conclusions

The three different software systems — PatBase, I2E and Intellixir — we have used in this work share some superficial similarity: they all offer tools for querying patent collections and displaying the resulting subsets. However, for the purpose of producing the patent landscape of the antibody–drug conjugates technology described in this paper, the differences between the programs are more important. Thus, PatBase has an extensive database of patent families seamlessly integrated in the software, allowing rapid selection of potentially relevant collections of patent publications. I2E has unmatched, NLP-based querying capabilities to precisely analyze text of such collections. Intellixir offers user-friendly, interactive visualization tools. Combining the best features of each of these programs produces results beyond the capabilities of any of them alone.

We did not intend to present in this paper a recipe for an optimal patent data processing scheme. Rather, we wanted to promote the concept that the ability to easily integrate multiple tools leads to optimal results in patent information analysis. The choice of programs used here reflects personal preferences, and there are products on the market which could be used to substitute for each of them. However, we are not aware of a single product that could support the entire workflow to the extent we described here.

The advantages of software integration are generally acknowledged. All the programs reviewed here offer the ability to export and/or import data in multiple formats compatible with third party applications. In addition, Linguamatics also provides some specific tools for data conversions: Perl scripts to convert various public ontologies into a standard format suitable for building I2E indexes, and Accelys’ Pipeline Pilot components to include I2E searches in automated workflows. The PatMOB program described in this paper goes beyond such solutions, offering high levels of customization and support for work with patents. In addition to one-click automated data conversions easily implemented by plug-ins, users can rapidly edit resulting patent collection, access the relevant information online, and store the results in an internal, searchable database.

As the volume and complexity of patent information increases, so does the complexity of software tools required to process that information. Software vendors keep adding new features, trying to satisfy customer demands for a one-stop solution. Integration tools like PatMOB could help to simplify this situation and promote innovation in the field, by allowing vendors to concentrate on their real strengths and encouraging users to “mix and match” resources for the optimal solution to their needs.

Acknowledgments

1. This article was developed based on a presentation made at the Patent Information Users Group (PIUG) 2012 Biotechnology Meeting in Boston, February 2012.

2. It should be noted that P.M., in addition to his current work at Sanoﬁ US, is also the founder of genemob.com and principal developer of the PatMOB software.

Annex. PatMOB patent software platform

PatMOB (Patent Mobile Objects) is a computer program designed to facilitate rapid development of plug-ins for management of various patent data formats. Written in the Java programming language, it runs on all major operating systems – Windows, Mac OS and Linux. It combines an intuitive graphical user interface (GUI), with an application programming interface (API) exposing a rich core infrastructure for plug-in development.

PatMOB plug-ins are Java programs ranging from small snippets written ad hoc to address the emerging needs – to full programs with their own business logic and complex GUI, seamlessly integrated with the growing functionality of PatMOB core infrastructure. This infrastructure includes:

- Relational database (JavaDB) for permanent storage of patent collections and notes in a tree structure
- Built-in support for fetching real-time patent information on the Internet, such as the full publication cycle bibliographic data from the Open Patent Services (APS) at the European Patent Office
- XML parsers
- Local file system access for reading and writing XML files, reports and various user data and preferences
- Tree Branch Editor which allows viewing and editing of patent collections – newly created, imported or retrieved from the internal database – as tree structures

These core resources can be accessed both programmatically by the plug-ins and interactively by the user. Thus, for example, a specialized plug-in can be used to import a patent collection from any available source and open it in the Tree Branch Editor. The user can then review, annotate and edit the collection in a familiar interface, and save it in the internal database. Later, another plug-in can be used to reformat and export any branch of the database tree for use in another application. PatMOB core infrastructure is undergoing active development — see the genemob.com website [22] for details and updates.

References


