Intelligent Information Access from Scientific Papers

Ted Briscoe[1,2], Karl Harrison[1], Andrew Naish[3],
Andy Parker[1], 3, Marek Rei[1], Advaith Siddharthan[4],
David Sinclair[3], Mark Slater[1], and Rebecca Watson[2]

[1]University of Cambridge, [2]iLexIR Ltd,
[3]Camtology Ltd, [4]University of Aberdeen

Ted.Briscoe|Marek.Rei@cl.cam.ac.uk,
Harrison|Slater|Parker@hep.phy.cam.ac.uk,
A.Naish@gmail.com, Advaith@abdn.ac.uk,
David.Sinclair@imense.co.uk, Bec.Watson@gmail.com

September 8, 2010

Abstract

We describe a novel search engine for scientific literature. The system allows for sentence-level search starting from portable document format (PDF) files, and integrates text and image search, thus facilitating the retrieval of information present in tables and figures. It allows the user to generate in an intuitive manner complex queries for search terms that are related through particular grammatical (and thus implicitly semantic) relations. The system uses grid processing to parallelise the analysis of large numbers of scientific papers. It is currently undergoing user evaluation, but we report some preliminary evaluation and comparison with Google Scholar, demonstrating its utility. Finally, we discuss future work and the potential and complimentarity of the system for patent search.

1 Introduction

Scientific, technological, engineering and medical (STEM) research is entering the so-called 4th Paradigm of "data-intensive scientific discovery" in

which advanced data mining and pattern discovery techniques need to be applied to vast datasets in order to drive further discoveries. A key component of this process is efficient search and exploitation of the huge repository of information that only exists in textual or visual form within the "bibliome", which itself continues to grow exponentially.

Today's computationally driven research methods have outgrown traditional methods of searching for scientific data creating a significant, widespread and unfulfilled need for advanced search and information extraction. Our system integrates text and image processing in order to create a unique solution to fine-grained search and information extraction for scientific papers. In this paper, we describe the current version of our system demonstrator focussing on its search capabilities.

We have developed a prototype search and information extraction system, which is currently undergoing usability testing with the curation team for FlyBase, a $1m/year NIH-funded curated database covering the functional genomics of the fruit fly. To provide a scalable solution capable, in principle, of analysing the entire STEM bibliome of around 20m electronic journal and conference papers, we have developed a distributable and robust system that can be used with a grid of computers running distributed job management software.

This system has been deployed and tested using a subset of the resources provided by the UK grid for Particle Physics [4], part of the worldwide grid assembled for the analysis of the petabyte-scale data volumes to be recorded each year by experiments at the Large Hadron Collider in Geneva. To build the current demonstrator we processed around 15k papers requiring about 8k hours of CPU time in about 3 days with up to 100 jobs running in parallel. A distributed spider for finding and collecting open access portable document format (PDF) versions of papers has also been developed. This has been run concurrently on over 2k cores, and has been used to retrieve over 1m subject-specific papers from a variety of STEM fields to date. However, the demonstrator, as discussed below, indexes about 10k papers on the functional genomics of the fruit fly.

2 Functionality

Our search and extraction engine is the first to integrate a full structural analysis of a scientific paper in PDF identifying headings, sections, captions and associated figures, citations and references with a sentence-by-sentence grammatical analysis of the text and direct content-based visual search over figures. Combining these capabilities allows us to transform paper search

from keyword-based paper retrieval, where the end result is a set of putatively relevant PDF files which need to be read, to information search and extraction, based on the ability to interactively specify a rich variety of linguistic patterns which return sentences in specific document locales and which combine text with image-based constraints – for instance:

> "all sentences in figure captions which contain any gene name as the theme of *express* where the figure is a picture of an eye"

The system allows the user to build up such complex queries quickly though an intuitive process of query refinement.

Figures often convey information crucial to the understanding of the content of a paper and are typically not available to search. Our search engine integrates text search to the figure and caption level with the ability to re-rank search returns on the basis of visual similarity to a chosen archetype (ambiguities in textual relevance are often resolved by visual appearance). Figure 1 provides a compact overview of the search functionality supported by the demonstrator. Interactively, constructing and running such complex queries takes a few seconds in our intuitive user interface, and allows the user to quickly browse and then aggregate information across the entire collection of papers indexed by the system. For instance, saving the search result from the example above would yield a computer-readable list of gene names involved in eye development in less than a second on a standard 64bit machine indexing around 10k papers. With existing web portals and keyword based selection of PDF files (for example, Google Scholar, ScienceDirect, Zotero or Mendeley), a query like this would typically take many hours to execute, requiring each PDF file returned to be opened and read in a PDF viewer, and cut and paste to extract relevant gene names. The only other current solution would require expensive customisation of a text mining / information extraction system by IT professionals using licensed software (such as that provided by Ariadne Genomics, Temis or Linguamatics). This option is only available to a tiny minority of researchers working for large well-funded corporations.

3 Summary of Technology

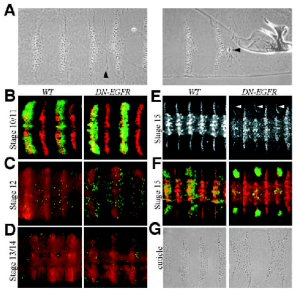
3.1 PDF to SciXML

PDF was developed to represent a document in a manner designed to facilitate printing. In short, it provides information on font and position for textual and graphical units. To enable information retrieval and extraction, we need to convert this ubiquitous typographic representation into a logical

one that reflects the structure of scientific documents ([2]). We use an XML schema called SciXML ([10]) that we extend to include images. We linearise the textual elements in the PDF, representing these as elements in XML and classify these divisions as {Title|Author|Affiliation|Abstract| Footnote|Caption| Heading|Citation|References|Text} in a constraint satisfaction framework.

In addition, we identify all graphics in the PDF, including lines and images. We then identify tables by looking for specific patterns of text and lines. A bounding box is identified for a table and an image is generated that overlays the text on the lines. Similarly we overlay text onto images that have been identified and define bounding boxes for figures. This representation allows us to retrieve figures and tables that consist of text and graphics. Once bounding boxes for tables or figures have been identified, we identify a one-one association between captions and boxes that minimises the total distance between captions and their associated figures or tables. The image is then referenced from the caption using a "SRC" attribute; for example, in (abbreviated for space constraints):

<CAPTION SRC= "FBrf0174566_fig_6_o.png"> Fig. 6. Phenotypic analysis of denticle belt fusions during embryogenesis. (A) The denticle belt fusion phenotype resulted in folds around the surrounding fused... ...(G) ...the only cuticle phenotype of the DN-EGFR-expressing embryos was strong denticle belt fusions in alternating parasegments (paired domains).</CAPTION>



Note how informative the caption is, and the value of being able to search this caption in conjunction with the corresponding image (also shown above).

3.2 Natural Language Processing

Every sentence or smaller textual unit, including those in abstracts, titles and captions, is run through our named-entity recogniser (NER) and syntactic parser. The output of these systems is then indexed, enabling more precise search.

Named Entity Recognition

NER in the biomedical domain was implemented as described in [11]. Gene Mention tagging was performed using a Conditional Random Fields model (using the MALLET toolkit [9]) and syntactic parsing (using RASP [3], using features derived from grammatical relations to augment part-of-speech (PoS) tagging. We also use a probabilistic model for resolution of non-pronominal anaphora in biomedical texts. The model focuses on biomedical entities and seeks to find the antecedents of anaphoric expressions, both coreferent and associative ones, and also to identify discourse-new expressions [5], and we deploy a reference parsing and citation linking module during the processing pipeline. The combination of these modules allows us to identify and distinguish mentions of author names, gene names, and gene products or components such as protein names, DNA sequence references, and so forth.

Both the NER and anaphora resolution modules of our processing pipeline are domain-specific. However, both are weakly-supervised and rely on extant ontologies or domain information, such as the gene names recorded in FlyBase, to generate training data and/or dictionaries. Therefore, these components are extensible to further scientific subfields for which similar ontologies and resources can be found.

Parsing

The RASP (Robust Accurate Statistical Parsing [3]) toolkit is used for sentence boundary detection, tokenisation, PoS tagging, morphological analysis and finding grammatical relations (GR) between words in the text. GRs are triplets consisting of a relation-type and two arguments and also encode morphology, word position and part-of-speech; for example, parsing "John likes Mary." gives us a subject relation and a direct object relation:

(|ncsubj| |like+s:2_VVZ| |John:1_NP1|)
(|dobj| |like+s:2_VVZ| |Mary:3_NP1|)

Representing a parse as a set of flat triplets allows us to index on grammatical relations [1], thus enabling more complex relational queries than is standard in scientific search engines.

The RASP system is relatively domain-independent compared to alternative statistical parsers. Lexical information is only used within the PoS tagger which also integrates a sophisticated unknown word handling module. The parser operates on PoS tag sequences and ranks alternative parses using structural information drawn from balanced training data. Nevertheless to improve handling of the large proportion of unknown words, we use

the predictions of the NER module to retag names with the correct PoS tag in cases where the tagger chooses an alternative, and we save the top five highest-ranked parses for indexing to improve recall in cases where the preferred parse is not correct.

3.3 Image Processing

We build a low-dimensional feature vector to summarise the content of each extracted image. Colour and intensity histograms are encoded in a short bit string which describes the image globally; this is concatenated with a description of the image derived from a wavelet decomposition [8] that captures finer-scale edge information. Efficient similar image search is achieved by projecting these feature vectors onto a small number of randomly-generated hyperplanes and using the signs of the projections as a key for locality-sensitive hashing [6].

Thus our current image similarity search is based on unsupervised clustering with some tuning of feature weights to achieve useful results in this domain. In the near future we will add supervised classifiers capable of recognising common subclasses of image occurring in papers, such as graphs, plots, photographs, etc., based on training data derived automatically via captions unambiguously identifying the accompanying image type.

3.4 Indexing and Search

We use Lucene [7] for indexing and retrieving sentences and images. Lucene is an open source indexing and information retrieval library that has been shown to scale up efficiently and handle large numbers of queries. We index using fields derived from word-lemmas, grammatical relations and named entities. At the same time, these complex representations are hidden from the user, who, as a first step, performs a simple keyword search; for example *express Vnd*. This returns all sentences that contain the words *express* and *Vnd* (search is on lemmatised words, so morphological variants of express will be retrieved). Different colours represent different types of biological entities and processes (green represents a gene), and blue words show the entered search terms in the result sentences an example sentence retrieved for the above query follows:

> It is possible that like ac , sc and l'sc , vnd is expressed initially in cell clusters and then restricted to single cells .

Next, the user can select specific words in the returned sentences to indirectly specify a relation. Clicking on a word will select it, indicated by underlining of the word. In the example above, the words *vnd* and *expressed*

have been selected by the user. This creates a new query that returns sentences where *vnd* is the subject of *express* and the clause is in passive voice. This retrieval is based on a sophisticated grammatical analysis of the text, and can retrieve sentences where the words in the relation are far apart; an example of a sentence retrieved for the refined query is shown below:

> First , vnd might be spatially regulated in a manner similar to ac and sc and selectively expressed in these clusters .

Once a user is confident that a ground pattern of this type is retrieving relations of interest appropriately, it is possible to 'wildcard' an argument of a predicate or abstract from a specific member of a semantic class, such as the gene *Vnd* to the entire class, in this case of genes. Figure 1 (step 3) shows a screenshot of the interface supporting this functionality

The current demonstrator offers two further functionalities. The user can browse the MeSH (Medical Subject Headings) ontology and retrieve papers relevant to a MeSH term. Also, for both search and MeSH browsing, retrieved papers are plotted on a world map; this is done by converting the affiliation of the first author into geospatial coordinates. The user can then directly access papers from a particular research group indexed with a specific MeSH term.

4 Evaluation

The demonstrator is currently undergoing user trials with members of the FlyBase curation team. They are faced with an increasing number of papers that they have identified as potentially curatable and downloaded on the basis of keyword search. The process of deciding whether a paper should be fully curated (approximately a person/day of effort), lightly curated recording, for example, genes mentioned, or ignored is itself time consuming and currently done by uploading a PDF to a viewer and/or printing it, and then reading it.

The system potentially speeds up this process by allowing a collection of papers to be searched at the sentence level for key phrases that indicate relevant information. For example, predicates such as *characteriz/se* with gene names as objects often indicate new information about a gene, whilst assignment of a mnemonic name to a sequenced gene denoted by a numerical identifier prefixed with *CG* is a good clue that a paper contains the first significant investigation of that gene. The ability to define patterns in the interface that find such characterisation or naming events from the text, means that, in principle, fully-curatable papers can be identified much more quickly.

Although it is too early to report on these usability experiments, we have conducted preliminary exploration of some common types of searches using intrinsic evaluation methods common in Information Retrieval, such as the (Mean) Average Precision measure. This is appropriate when we are evaluating a system that ranks sentences according to a given query where we want to measure the degree to which relevant sentences are ranked higher than irrelevant sentences and all relevant sentences appear in the ranking. A single query version of average precision is defined by:

$$\frac{\sum_{r=1}^{N}(Prec(r) \times TP?(r))}{TruePositives + FalseNegatives} \tag{1}$$

where N is the number of sentences returned by the system, r is the rank of the sentence, and $TP?$ returns one (zero) if the r^{th} sentence is (not) a true positive and Prec(ision) is defined as:

$$\frac{TruePositives}{TruePositives + FalsePositives} \tag{2}$$

so a score of one entails perfect recall and ranking.

We start by considering a relatively simple goal like 'find all sentences which discuss Adh expression in fruit flies' where Adh is a gene name and we are interested in expression events with Adh as theme. As illustrated in section 3.4, keyword search can be refined to enforce the appropriate semantic relation between the gene name and some form of the predicate *express*, and near synonyms such as *overexpress* if desired. The goal then is to retrieve sentences containing phrases like a), b) or c) below, but not d).

a. ...express Adh...

b. ...expression of Adh...

c. Adh is one of the most highly expressed genes...

d. Adf-1 is an activator of Adh that was subsequently shown to control expression of several Drosophila genes...

Our system allows the user to achieve this by constructing a (disjunctive) set of queries which define various appropriate grammatical patterns, Note that standard IR and search engine refinements like string search or operators like NEAR cannot achieve the same effect. The former achieving high precision but low recall, the latter achieving a better approximate ranking, but not directly enforcing grammatical / semantic constraints.

To achieve this goal using Google Scholar (or any other document-level search system such as those offerred by the major scientific publishers, academic associations, etc.) a sophisticated user might construct the following query:
(Adh OR alcohol dehydrogenase OR CG32954) NEAR (expression OR express OR overexpress) AND Drosophila
This yielded about 15k papers together with header and text snippets (in July 2010). Using the headers and snippets, the user now has to decide whether to save a PDF for further investigation or not. The information available before downloading and opening the paper in a PDF viewer is sometimes adequate to accept or reject a paper, but also often unclear. For example, the snippet in a) below clearly shows this paper contains a relevant sentence; that in b) strongly suggests the paper contains no relevant sentence, but that in c) is unclear because the first snippet has been truncated after *the* so the critical information is missing.

a. Identification of cisregulatory elements required for larval expression of the Drosophila melanc-gaster alcohol dehydrogenase gene. ...

b. Hypomorphic and hypermorphic mutations affecting the expression of Hairless. ... The genetics of a small autosomal region of Drosophila melanogaster, including the structural gene for alcohol dehydrogenase.

c. The Molecular Evolution of the Alcohol Dehydrogenase and Alcohol Dehy-drogenase-related Genes in the ... The DNA sequences of the A&z genes of three members of the Drosophila melanogaster species ...

Furthermore, the ranking of papers given by Google Scholar does not ensure that clearly relevant snippets occur before unclear or irrelevant ones, as ranking is based on a combination of the frequency of keyword occurrences through the paper and on keyword density within snippets. For example, b) above occurs before c), whilst the 50th page of results still contains three (out of ten) papers with clearly relevant snippets and the 99th page one clearly relevant snippet. Indeed, after the first few pages where most snippets are clearly relevant, the ranking 'flattens' so that most pages sampled throughout the set returned contain one or two clearly relevant snippets.

We estimate that a comprehensive search of papers with relevant snippets would involve downloading and viewing about 1K papers, though even then there would be little hope of achieving full recall, given the unclear status of a significant number of headers and snippets. For each paper downloaded, a PDF viewer's Find feature can be used to quickly move to potentially

Query	1	2	3	4
	0.735	0.758	0.855	0.933

Table 1: Average Precision for *Adh* as theme of *express*

relevant sentences. We sampled 10 papers with relevant snippets and found that in general *express* was the more restrictive keyword. On average, we found about 100 matching sentences of which about 10 exhibited the relevant relationship, whilst it took about 10 minutes per paper to identify these sentences. A conservative estimate of the time taken to identify the entire set of relevant sentences in papers clearly identified as relevant by Google Scholar would be about one month. The average precision of this approach – assuming that relevant sentences within papers are uniformly distributed, factoring in snippet identification, but assuming full recall via clearly relevant snippets – would be about 0.1 over the first 30 or so papers and about 0.001 over the full set. In a sense this analysis is unfair as Google Scholar is designed to be a paper retrieval system. Nevertheless, it is probably the best generally-available tool for the task today, as the snippet information surpasses anything provided by other scientific paper search sites, such as Elsevier's ScienceDirect, and its coverage of the literature is unrivalled.

To estimate performance in our demonstrator we used the Lucene command-line query language back-end to retrieve all sentences which contained a form of *express* or one of its near synonyms and *Adh* or one of its synonyms. We then manually classified this set of sentences into those which were relevant or not, and used this gold standard to compute average precision scores for four variant queries. Query 1 simply used the ranking obtained searching for *Adh* and *express* in the same sentence, query 2 required some path of grammatical relations linking these two keywords, query 3 added synonyms for each keyword, and query 4 enforced some path of grammatical relations between each set of synonyms and scored the sentences for ranking according to the length of this path to favour shorter paths. The average precision for each of these queries is given in Table 1. Gains in precision of several orders of magnitude are made over using Google Scholar and a PDF viewer, simply by supporting (Boolean) keyword search over sentences and returning these rather than PDFs. However, grammatical constraints also yield a significant improvement in the overall ranking obtained, effectively ensuring that, for the first two pages of results returned, all sentences are relevant.

So far, we have only considered searches involving ground terms, but the system allows search via patterns over semantic / named entity classes

Query	1	2	3	4	5
	0.116	0.461	0.552	0.512	0.562

Table 2: Average Precision for *CG* naming events

or partially wildcarded terms. As mentioned above, curators would like to find papers that contain naming events involving *CG* prefixed identifiers, as these are a useful clue that a paper should be fully curated with respect to the named gene. We used the Lucene query language to find sentences containing variants of the predicate *name (X Y)* and synonyms like *call (X Y)*, *refer (to X as Y)*, etc along with any lemma matching *CG** and then manually classified the resulting set to identify relevant sentences containing a naming event between the *CG* identifier and a gene name. We then used this gold standard to compute average precision for 5 variant queries. Query 1 simply searched for sentences containing *CG** and a variant of *name*, query 2 added synonyms of *name* as above, query 3 disjunctively specified a set of known patterns that picked out grammatical constructions likely to specify a naming relation, like 'CGID *referred to as* GENE' or 'CGID (GENE)', query 4 allowed any path of grammatical relations between the *CG* identifier and a naming predicate scored by length, and query 5 combined the specific grammatical patterns (query 3) and the general path constraint (query 4). The average precision for each of these queries is given in Table 2. In the case of this more complex relational query between classes of terms, overall performance is poorer but the differential advantage of enforcing grammatical constraints is also much greater in this case than a simple requirement for cooccurrence of terms within a sentence.

The current user interface doesn't support the general path constraint on grammatical relations. Therefore, curators need to disjunctively specify a range of grammatical patterns and collate the results of each of these manually. We are redesigning the system to support automatic expansion of queries to add semantically-equivalent grammatical patterns and to enforce the path constraint by default in refined searches specifying any grammatical constraint. For instance, returning to the example in section 3.4, a user who selects *express* and *Vnd* in a sentence where *Vnd* is the subject of the passive verb group *is expressed* would automatically be shown further sentences in which *Vnd* is object of an active or nominalised form of the verb, such as *expressed Vnd* or *expression of Vnd*, and sentences in which any path of grammatical relations between a form of *express* and *Vnd* is found, such as *expression of ac and sc often with Vnd*, would be returned, albeit with lower

ranking.

The ranking of search results yielded by complex queries with multiple constraints on images and text is sometimes unintuitive, as are the results of similarity-based image search. We are adding classification to the image search, exploiting caption information to gather labelled training data, so that results hopefully will be less arbitrary than those sometimes achieved by unsupervised clustering. We are moving to a faceted, Boolean model of query constraint integration so that scoring and ranking of results will play a less central role in 'navigation' towards a satisfactory query formulation.

Nevertheless, even using the current interface it is possible to identify sets of papers, using queries of this type, for full curation with satisfactory recall and good enough precision. This process takes less than an hour rather than the weeks required to achieve similar ends using other widely-available scientific paper search systems.

5 Conclusions and Further Work

To our knowledge, this is the first time that content-based image and advanced text processing have been integrated to provide fine-grained search over scientific papers. Our preliminary experiments suggest that the resulting system has the potential to greatly improve search and information extraction with complex documents. In order to develop the system in a scalable and relatively domain-independent fashion, we have utilised the grid and distributed processing to spider and annotate papers, and weakly-supervised machine learning methods or domain-independent modules in the annotation process. Our annotation pipeline is the first developed which is able to preprocess a PDF, identify the internal structure, and represent the result in a manner which supports application of state-of-the-art image and text processing techniques.

Nevertheless, there is much work to be done before all of the our aims are achieved. Firstly, we need to demonstrate that the weakly-supervised NER and anaphora resolution modules can be ported effectively to new (sub-)domains or that they can be replaced without serious loss of search performance by unsupervised techniques. Secondly, we need to evaluate the user interface with a wider group of potential users and to explore and develop its effectiveness for other fields, such as computer science, which differ from genetics in terms of the likely focus of searches. Thirdly, we nned to extend system functionality and the interface to support information extraction. This will require the ability to save and reapply complex queries once they have been developed incrementally and interactively to a point

where the user is satisfied with their performance. Where these (relational) queries match classes of terms, it would also be useful to be able to save the lists of ground terms that match in a computable-readable format and also to re-use such lists during the formulation of further queries.

The system is potentially relevant to patent search professionals for several reasons. Firstly, we believe that the techniques we have developed for search and information extraction from scientific papers are broadly applicable to any collection of relatively complex documents containing technical terminology, images, and internal structure, such as patents. In addition, our current demonstrator also supports access to papers via the MeSH ontology, and this could be straightforwardly extended to support access to patents via any of the ontologies developed to support patent search. Secondly, there are many similarities between patent and scientific paper search which demarcate both from general web search. Both often involve fine-grained and comprehensive search for information rather than keyword-based access to a document or page ranked by popularity or frequency of keywords. And both are conducted by professionals willing to develop 'advanced search' expertise whose search sessions typically last hours rather than minutes. Finally, patent searchers are frequently interested in prior art and prior art can potentially be found in the scientific bibliome. In the longer run, combined search over both patents and scientific papers using the same interface and search tools would be very valuable.

6 Acknowledgements

This work was supported in part by a BBSRC e-Science programme grant to the University of Cambridge (FlySlip), and a STFC miniPIPSS grant to the University of Cambridge and iLexIR Ltd (Scalable and Robust Grid-based Text Mining of Scientific Papers). This paper is an extended version of one which appeared in the proceedings of the annual North American Association for Computational Linguistics conference proceedings, demonstration session, in June 2010.

References

[1] Michaela Atterer and Hinrich Schutze. An inverted index for storing and retrieving grammatical dependencies. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008.

[2] Dan Tidhar Bill Hollingsworth, Ian Lewin. Retrieving hierarchical text structure from typeset scientific articlesa prerequisite for e-science text mining. In *Proceedings of the 4th UK E-science All Hands Conference*, pages 267–273, Nottingham, UK, 2005.

[3] Ted Briscoe, John Carroll, and Rebecca Watson. The second release of the rasp system. In *Proceedings of the COLING/ACL 2006*, Sydney, Australia, 2006.

[4] D. Britton, AJ Cass, PEL Clarke, J. Coles, DJ Colling, AT Doyle, NI Geddes, JC Gordon, RWL Jones, DP Kelsey, et al. GridPP: the UK grid for particle physics. *Philosophical Transactions A*, 367(1897):2447, 2009.

[5] C. Gasperin and T. Briscoe. Statistical anaphora resolution in biomedical texts. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 257–264, 2008.

[6] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *In Proc. 25th Internat. Conf. on Very Large Data Bases*, 1999.

[7] B. Goetz. The Lucene search engine: Powerful, flexible, and free. *Javaworld http://www. javaworld. com/javaworld/jw-09-2000/jw-0915-lucene. html*, 2002.

[8] C.E. Jacobs, A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277–286. ACM New York, NY, USA, 1995.

[9] McCallum A. K. Mallet: A machine learning for language toolkit. *http://mallet.cs.umass.edu*, 2002.

[10] Simone Teufel, Jean Carletta, and Marc Moens. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 110–117, 1999.

[11] A. Vlachos. Tackling the BioCreative2 gene mention task with conditional random fields and syntactic parsing. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 2007.

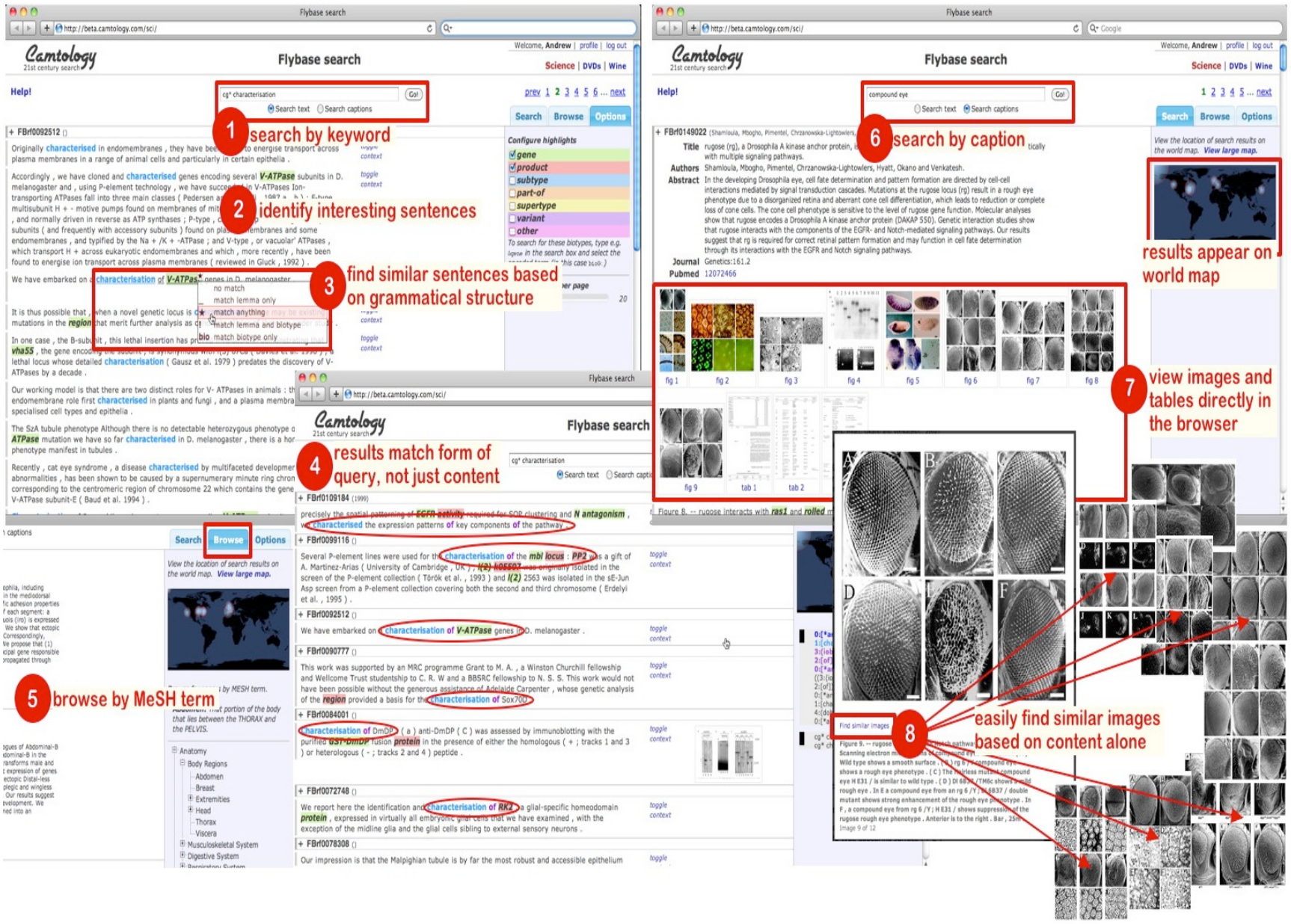


Figure 1: Screenshots showing functionality of our demonstrator