SOA-based Integration of Text Mining Services

Johannes Starlinger[1], Florian Leitner[2], Alfonso Valencia[2], Ulf Leser[1]
[1]*Humboldt-Universität zu Berlin, Berlin, Germany*
[2]*Structural Biology and BioComputing Programme, CNIO, Madrid, Spain*
{starling,leser}@informatik.hu-berlin.de, {fleitner,valencia}@cnio.es

Abstract

Text Mining has established itself as a valuable tool for knowledge extraction in many commercial and scientific areas. Accordingly, a large number of different methods have been developed focusing on a broad range of different tasks. We report on a novel system architecture that is fundamentally service-based, i.e., it models and implements text mining and knowledge extraction routines as independent, yet federated services. The system has several layers: (1) Base services perform various fundamental extraction tasks. They all implement a fixed interface but keep their particular algorithms and functionality. (2) A metaservice acting as a central access point to those base services, thus providing a homogeneous interface to different algorithms. (3) An aggregation service on top of the metaservice which implements functionality to graphically show, compare, and aggregate the results of different base services. Each layer is accessible as a Web Service and thus ready to be integrated in applications that are higher up in the value chain, such as authoring tools or systems for the automatic construction of knowledge bases. We developed our system with a focus on the mining of Life Science text collections. It is available from http://www.bc-viscon.net.

1. Problem Background

Much of the present knowledge in the Life Sciences (as in many other domains) is not available in structured form but only in scientific articles [4]. With the steeply increasing levels of data production, it has become infeasible that a human researcher keeps up to date with the latest findings even in very small areas. Therefore, automatic methods for finding and extracting knowledge from large collections of articles, written in natural language, are needed.

Text mining encompasses a set of techniques that target this task [16]. A particular relevant technique is information extraction, which denotes algorithms and methods for finding and extracting specific pieces of information from unstructured text. Information extraction typically is performed in the form of analysis pipelines, where several algorithms are called one after the other such that the output of each step in the pipeline is the input to the next step. Typical steps are format conversion, sentence splitting, tokenization, word stemming or lemmatization, annotation of tokens with their part-of-speech (adjective, verb, article etc.), deep or shallow parsing, recognition of semantic units such as names (of persons, organizations, products etc.), dates, location etc., and the detection of relationships between entities. For each of these steps, various algorithms have been proposed and implemented. They all have their particular strengths, depending on the type of text, the type of entities searched for, the domain etc.

The human organism has approximately 25000 genes, each of which is important for one or more particular functions. Many genes for decades have been and still are the target of intensive research. For instance, p53, an important gene in the onset of many cancer types, is mentioned in almost 50000 publications; MCY, a gene involved in regulation of cell cycle, is mentioned in app. 20000 articles. Such well-studied genes are also well represented in structured databases. On the contrary, for almost 50% of the human genes no structured functional information is available. Although researchers had studied many of them, they reported on their findings only in papers. At the same time, complex and endemic diseases such as diabetes, cardio-vascular diseases, rheumatic diseases etc., are associated to dozens or sometimes even hundreds of genes, the function of which often is only described in text.

Text mining has established itself as an indispensable tool to get an overview of the genetic background of such diseases. The most important tasks that need to be solved are (a) recognition of gene and protein names (named entity recognition, NER), (b) recognition of functional information, and (c) recognition of the relationship between biological entities (such as protein-protein interaction or gene-gene regulation) [7]. For each of these steps, various algorithms have been proposed and implemented (see [14] or [18] for surveys). Their performance differs largely depending on the particular task at hand. For instance, the accuracy of tools for recognition of gene names depends to a great extent on the species that has the gene. While genes of humans, mice or flies in general are very hard to detect (F-measure around 80%), yeast genes are much simpler (~90%) [10]. Tools for recognizing protein interactions differ in their performance depending on the structure of the texts, the precise definition of “interactions”, and many other

factors. Furthermore, all tools are tuned either towards precision or recall; finally, they offer different added services, such as the mapping of recognized objects to real-world entities (named entity normalization).

Accordingly, there are no one-suits-all text-mining tools, and particular projects always need to build their individual analysis pipelines. However, when we fix a particular task (such as NER for genes), the interfaces of all relevant tools can be described quite succinctly, as they basically all implement the same functionality and produce the same output (though in different formats). Therefore, it should be possible to construct systems where each tool developer implements its particular service in whatever form, but with a standard interface; building an entire text mining pipeline should then only be a matter of "gluing" together a selection of such services.

In this paper, we report on a system that implements this idea. We built a multi-layer application for the recognition of gene and protein names in scientific articles using a flexible composition of a set of independently developed and maintained services. The process is divided in three tasks (see Fig. 1):

- Given an arbitrary text, *base services* (BC-ASs – BioCreative Annotation Servers) recognize genes, proteins, taxa, and interactions in this text using their favorite algorithm. All base services implement the same interface and provide their results in a standard format. In the current system, there are twelve such base services running at institutions all over the world.
- A *metaservice* (BCMS – BioCreative MetaServer) controls and orchestrates a set of base services. The BCMS may be queried directly by providing a text, database identifier, or a reference to PubMed, the largest existing collection of abstracts in the Life Science[1]. It passes this information to the base services, gathers their results and integrates them into a uniform output. BCMS also maintains a cache of analyzed results for faster access and provides metadata on the base services and their statuses. Currently, it is ran and maintained in Madrid.
- An *aggregator service* (BC-VisCon) semantically integrates the results gathered by BCMS. To this end, it implements various routines to merge conflicting results (which happen when different base services annotate different regions in the text as genes) and to graphically analyze and compare results. VisCon can be used directly through a flexible Web2.0 interface, or may be called as a parameterized service from other applications. This service is currently set up in Berlin.

The system we present is, to our knowledge, the first application connecting distributed web services for text mining. It fundamentally builds on SOA principles and technology and demonstrates their use for scientific knowledge management in an important area, i.e., the Life Sciences. Research in the Life Sciences has since long been a worldwide distributed effort, with thousands of groups freely contributing data and programs. The immense flood of data can only be analyzed by a concerted effort of groups around the world [2]. We believe that our endeavor is a blueprint for setting up analysis pipelines using SOA technology.

The rest of this paper is structured as follows. In the following two subsections we detail the functionality of BCMS and BC-VisCon. An overview of their technical implementation is given in Section 2. We highlight the use of SOA and Web2.0 techniques in Section 3. Applications of our service pipeline are highlighted in Section 4. In Section 5, we conclude and discuss future extensions to our system.

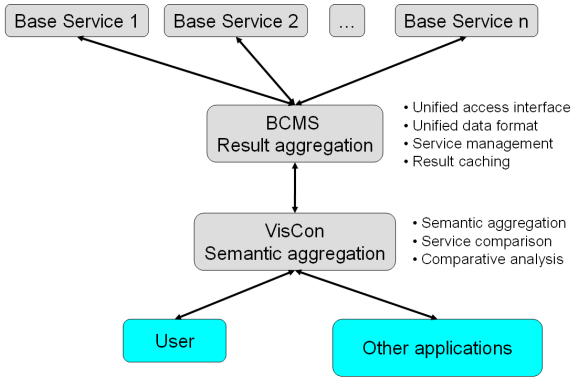

Figure 1: Interplay of base services, the BCMS metaservice, and VisCon. All services were developed independently and are hosted by different organizations.

1.1. BCMS, the BioCreative MetaServer

The BioCreative MetaServer (BCMS) [12] is a text-mining platform that was built following the blueprint of similar and highly successful result aggregation platforms in Bioinformatics, such as structure prediction services [5]. The common principle is unifying results from various algorithms running on the same task but in a distributed manner, thereby providing a collated view on the data that enables the service's user to directly compare and access different results.

BCMS started after the BioCreative II workshop, in which results on an international competition in biological text mining had been presented [11]. At this workshop, a large and international group of researchers decided that providing a standard interface to different annotations on biomedical texts would provide a great benefit to biomedical research. For text mining researchers, the BCMS provides a consensus for annotations, which can be exchanged, compared and reused throughout the community. For the administrator of bioinformatics

[1]PubMed currently indexes app. 16 Million abstracts and grows at a rate of app. 500.000 abstracts per year.

resources (databases etc.), the platform can provide added value to their existing systems (e.g., mapping protein records to Medline abstracts). Finally, a biomedical end-user can use it to have simple access to high-end text mining solutions. All these use cases are possible due to the SOA-based design of the platform, as demonstrated in this publication.

BCMS[2] provides four semantic annotation types for scientific texts (see below for details). These are (a) gene/protein names, (b) gene/protein normalizations, (c) biological taxa, and (d) protein-protein interactions. Currently, there are twelve annotation services connected to the platform. Through its SOA-based design principles, any number of additional Annotation Servers and annotation types could be integrated into the platform without changing the public XML-RPC interface, thus shielding other applications – such as BC-VisCon – from changes. The overall BCMS design is visualized in Fig. 2.

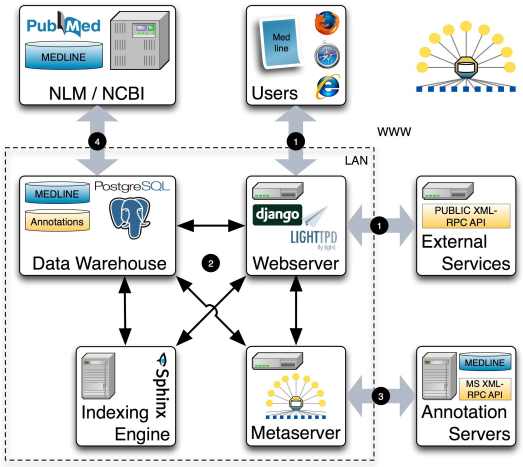

Figure 2: BCMS Architecture. (1) The web server listens for incoming HTTP or XML-RPC (such as BC-VisCon) requests. (3) Requests are forwarded to the BC-ASs. (2, 4) Results are enriched by external services and stored and indexed locally in a warehouse.

1.2. BC-VisCon, a Semantic Aggregator

The base services orchestrated by the BCMS provide some of the best NER systems for gene and protein names to date. The BCMS makes accessing their results simple; however, a user still has to chose which service might suit his needs best. This is not an easy task, as no easily accessible method existed for comparing annotation server results on a given text. VisCon provides this functionality. It is a third level application on top of the services provided by the BCMS. It takes a PubMed query, calls the PubMed query service to find matching abstracts, passes these (or a user-chosen subset) to the BCMS, and receives the annotations of the different annotation servers on all texts. As NER for gene names is not an easy task, the different results often are conflicting, i.e., different servers annotate different regions in the text as gene. VisCon provides services to (a) graphically compare the conflicting annotations (see Fig. 3), (b) compute a unified result using different consensus finding methods (such as union, intersect, majority etc.), and to (c) compute a hierarchical clustering of all services on a given subset of the texts to identify groups of services performing similar annotations. These functions are available through a Web2.0 interface and as XML-RPC services.

Note that this task is necessary because there is no ground truth in NER for genes. [10]. This starts with the question what a gene is – a question biologists are far from agreeing about – and ends with the unclear definition of what makes a gene's full name. For instance, the longest gene name we are aware of consists of not less than ten tokens: “human T cell leukemia lymphotropic virus type 1 Tax protein “; however, where words such as “human” or “protein” should be part of this gene name is a question of debate even among experts. It follows that users seeking annotations for their texts have a more or less precise, but in any case non-universal sense of the 'correctness' of a tagging. Any system for aggregating annotations thus cannot claim to be able to derive the “true” annotations, but can only provide tools for helping users to find “their” tagger – be it a single annotation server or be it the combination of several such servers.

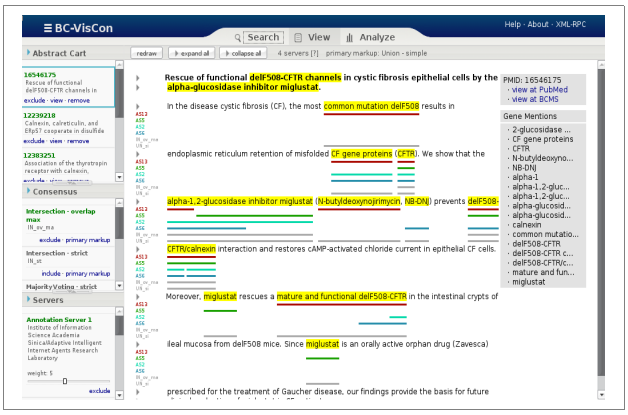

Figure 3: The BC-VisCon web interface showing contradicting annotations by different base services (colored lines) on the same text (text on top of lines). The three-paneled area on the left contain configuration options.

2. Technical Explanation

In this section, we briefly describe the techniques that were used to implement our distributed text mining pipeline.

2.1. BCMS

From the very start, the BCMS was designed on a service-based paradigm. This implies that all functionality is provided by services which are implemented independently and which are maintained worldwide. Prior

[2] The current prototype provides access to the corpus of approx. 22800 Medline abstracts used during BioCreative II.

to its creation, there actually was little incentive for text mining developers to provide their algorithms through web services; however, with the advent of the BCMS, many providers have wrapped their algorithms into a web service implementing the BCMS standard AS interface. Thus, the fundamental task of the BCMS service is that of integration.

The XML-RPC protocol was chosen as message exchange format because it is the most widespread and lightweight protocol available. We disregarded SOAP, even though it provides more functionality, because these advanced features are not required for the BCMS platform. REST was discarded, as there is no *standardized* way of sending large quantities of data from the client to the server[3], which is of importance for future versions when clients will be able to send arbitrary text to the meta-service.

The platform listens to client requests for a given Medline identifier. Requests are directly forwarded to the various Annotation Servers by creating client-specific threads. After the BC-ASs have completed their analysis, the results are received and validated by the BCMS. Validation is important to ensure data integrity. It encompasses several tasks: (a) ensuring that reported database identifiers map to existing records, (b) retrieval of the names for referenced identifiers, and (c) determining if a reported gene/protein name exists at the reported offset in the abstract. The first two tasks are implemented using external service calls to important databases.

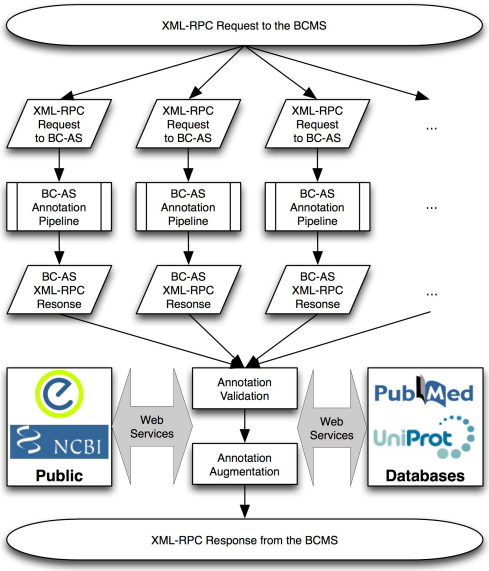

Figure 4: BCMS Process. The BCMS orchestrates incoming requests (stored in a priority queue) to the Annotation Servers. Returned results are validated and augmented with additional data.

[3] Standard-conform REST request are made via the URL parameter of a TCP/IP package, which is limited in number of characters.

Failed validations are discarded and reported to the system's administrator for review, while valid results are subsequently stored in the platform's database (see Fig. 4). Storing the results in a database allows the BCMS to respond quickly to re-occurring requests: instead of repetitively requesting annotations for a given text resource, the BCMS can directly return annotations from the cache. The annotation results themselves are modeled as simple as possible, with the intent to (a) reduce data transfer size, (b) streamline the data structure to minimal requirements, and (c) avoid conflict of interests (e.g., because of distribution limitations for Medline abstracts). For each annotation type a list of associative arrays (hash, dictionary) is returned with the following elements:

- Gene/protein names: the name string itself, character offset determining the string's position in the text, and a section designator ("title" or "abstract" for Medline),
- Gene/protein normalizations: a unique database identifier and the database' name,
- Biological taxa: the NCBI taxonomic ID and a confidence score, and
- Protein-protein interactions: array containing classification result (true/false) together with confidences scores.

For requests made to the BCMS, these arrays are grouped by Annotation Server and forwarded to the client. Optionally, the raw results can also be viewed in a web browser [12].

2.2. BC-VisCon

The central component of the VisCon web service is a server application build on top of the Catalyst MVC framework[4]. This framework allows for a completely modularized development of the components of the system (see Fig. 5).

The controller component of the system has three main duties: (a) finding and retrieving PubMed abstracts based on a users query, (b) finding the consensus of the annotation servers through various methods and (c) calculating analysis data for pair-wise comparison of annotation servers. It uses a number of modules for these tasks:

1. The *BCMS communications module* implements a convenient interface to the web service provided by the BCMS using the RPC::XML CPAN module. Through this model not only general information regarding the specifics of the annotation servers and the teams that deploy them, but also the annotations for specific abstracts are fetched.
2. The *PubMed module* allows searching for abstracts by id or keyword search using the WWW::Search::PubMed module providing access

[4] See http://www.catalystframework.org/

to the NCBI. This retrieval of abstracts directly from PubMed is necessary due to legal restrictions concerning the dissemination of abstracts. BC-VisCon only uses the abstracts for display in its user interface.

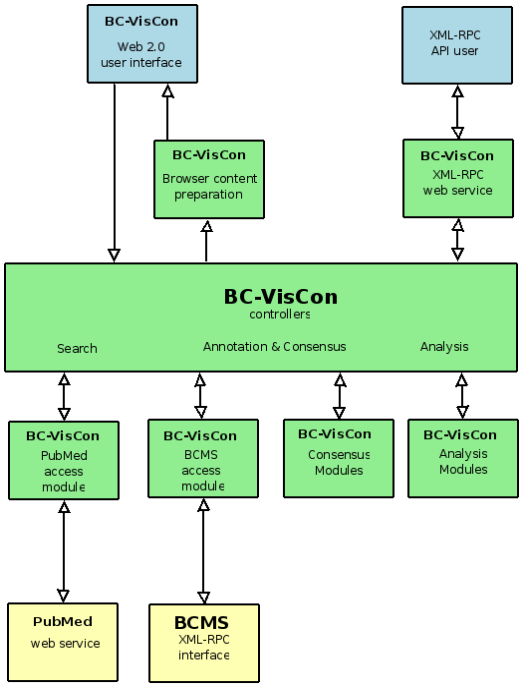

Figure 5: Schema of the BC-VisCon system architecture

3. The *consensus finding back-end* was designed in a highly modular way: Each consensus finding subunit (e.g. union, majority voting, etc) has its own model class providing one or more evaluation type denoting methods (e.g. strict, loose, etc.). For greatest flexibility and extensibility each of these classes has to register itself with a core module that keeps a central directory. This directory is later queried by the controllers. This approach makes it very easy to add new consensus finding methods. Furthermore, the internals of the subunits themselves are completely abstracted. It is also possible to add modules that engage external services for finding their consensus. To facilitate module development, the core consensus module provides various methods for common tasks, i.e. filtering of given annotations or grouping of several consensus finding methods.
4. The *analysis back-end* provides, for a given pair of annotation sets on one or more abstracts, methods for calculating the similarity of the annotations. To this end, all annotations are compared pair-wise. The results are stored in a base services similarity matrix, which is the basis for a hierarchical clustering algorithm to graphically show groups of similar services in a dendrogram. Another service of this module calculates precision, recall and f-measure of all servers wrt. to a selected server, which takes the role of a gold standard. Comparisons can be based on five different evaluation types: strict match, overlap, right-, left- and either-boundary-match. The evaluation is performed using micro- or macro averaging.

Founding on this solid base of model back-ends the controller component not only interconnects the models but also and foremost services their capabilities to the user. According to the three main duties described above and reflecting the underlying model layer architecture, this central component is divided into three main parts:

1. The *search unit* engages the corresponding method of the PubMed module to find abstracts and to retrieve the result list.
2. The *annotation unit* communicates with the BCMS for annotations and computes the consensus annotations. It accepts a number of parameters for fine-tuning, such as selecting specific BCMS-connected annotation servers or selecting the consensus-finding methods.
3. The *analysis unit* is responsible for computing similarities between annotation services. As opposed to the annotation unit it operates on multiple annotated abstracts, but takes the same set of tuning parameters. The flow of control is as follows: For a given list of PMIDs, annotations are loaded via BCMS and optionally filtered by the server selection parameter. The chosen consensuses are calculated and are added to the list of annotations. Finally, all resulting annotations are compared against each other for similarity and/or f-measure computation.

The clustering is also implemented in the analysis unit using the open source C Clustering Library[5], the output of which is reordered to form a list representation of a binary tree. This tree is visualized as a dendrogram using XHMTL and CSS (see Fig. 6).

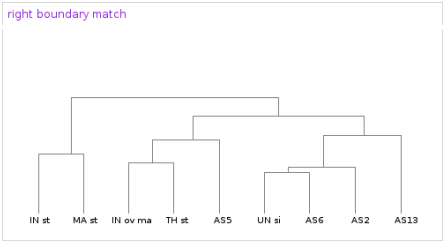

Figure 6: Dendrogram visualizing the similarity of different annotation servers and consensus methods for a right-boundary-match.

BC-VisCon offers access to its system in two ways. A web interface provides easy access to all features and also visualizes results. It lets the users interactively explore

[5] http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/software.htm

annotations made by the BCMS-connected servers in various ways using Web 2.0 paradigms and AJAX.

The second door to the system is an XML-RPC based web service offering all functionality (except the visualization) and that can be used by knowledge management applications further up the value chain (see Section 7). A full description of this interface can be found at http://www.bc-viscon.net/xmlrpc.

3. SOA Methodology

Our application is fundamentally built upon SOA principles. It emphasizes the aspect of integrating independently developed and distributed services into composed services that each offers an added value. The ensemble of BC-AS base services, the BCMS platform, and the BC-VisCon aggregator realizes a novel type of *service orchestration* in biomedical research benefiting from the primary design principle of SOA: *interoperability*. The BCMS acts as data provider encapsulating the BC-ASs, i.e., the autonomous services from the Annotation Servers are managed and presented by the BCMS. On the other side, the BC-VisCon service implements a fine-grained visualization on the resource (i.e., *abstraction* and *composition*). Finally, the general platform's *loose coupling* minimizes the metadata requirements that otherwise add unnecessary overhead to such frameworks. Another SOA paradigm, *federation of resources*, is intrinsic to our system in various aspects (see Fig. 7).

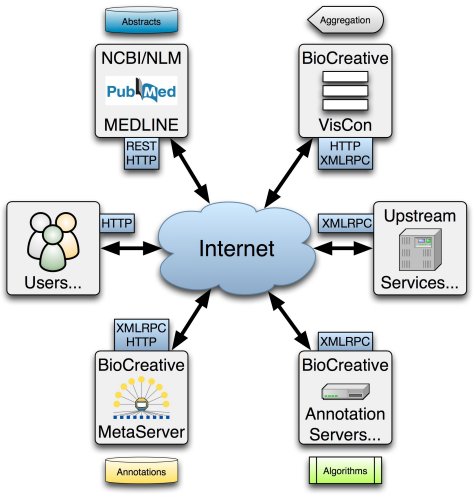

Figure 7: SOA Design. Each component is an independent entity with a defined interface (blue boxes). This underlines the loose coupling of each service: they are aware of each other, but can also act autonomously. The type of annotations provided by and the number of Annotation Servers is unlimited, as is the number of downstream services, which can be connected to any of these components. These two issues are regulated by the contract agreement (annotation types, message protocol) and enable individualized composition.

This ease of integration comes at the prize of a small reduction of the autonomy of the participating services: they must implement a common interface, and they must provide their data in a predefined format. However, these restrictions are little compared to the gains, which is proven by the high number of participating base services. Note that, before the creation of BCMS, only two gene NER applications were available for online use, while our system currently makes available 13 (not all of which are online all the time). An alternative would be to switch to a more loosely coupled implementation, where arbitrary services are integrated by means of automatic methods for service discovery and parameter matching [15], possibly using semantic web technology [3]. We closely follow the developments in these areas, but believe that current technology is not yet mature enough to offer automatic and high quality service integration.

A particular concept of our architecture is that of chaining services into pipelines. Each step in the pipeline performs a certain task and can be implemented by different services. Currently, only base services exist more than once, but we also envision that new providers will offer alternative implementations for the other steps. Such alternatives would immediately benefit from the developed architecture. Clearly, one could also break the complete pipeline into finer-grained steps. For instance, there could be proper services for tokenizing or linguistically annotating texts, prior to NER services making use of this data. Currently, we did not follow this approach due to the fact that such services are not offered (yet); and because they are not offered, these tasks usually are tightly integrated into the implementations of NER services. However, we are convinced that service providers will slowly start to break up their implementation and to offer interfaces for other services to hook in. Currently, demanding such a change in implementation would probably compromise autonomy of service providers too much.

BC-VisCon is not only available as XML-RPC but also acts as a user portal to the BCMS and BC-VisCon functionality. This portal makes extensive use of Web2.0 technologies that allowed us to create a comfortable and full-fledged tool for online usage, providing informative and appealing visualizations. Besides a track-enabled view of the annotations made by the back-end servers and consensus methods (see Fig. 3), the comparisons of annotations are available as similarity matrices and as an intuitive dendrogram (see Fig. 6). A control panel lets the user choose which abstracts, consensus methods and annotation servers to use.

BC-VisCon has not yet been optimized for performance. The time it takes to create an answer to the various services depends on a variety of factors, such as the speed of the base annotators, the number of recognized gene names in the abstracts, the number of abstracts which a comparison is based upon etc. Only some of these

factors can be influenced in the BCMS or in BC-VisCon, and we paid attention to do so as much as possible, using, for instance, parallel AS calling and caching of results. However, consensus results can not be reasonably cached due to the countless number of combinations of consensus methods, abstracts, and servers included in consensus finding.

4. Innovation

The project we present in this paper is the first system world-wide that leverages the advantages of a service-based approach to the construction of distributed systems for text mining, an essential component of current knowledge management systems. Component-based systems for text mining have been developed before (such as LingPipe[6] or UIMA [8]), but none of them is as simple to use, has so few requirements for tool developers to participate, and was from scratch designed for the distributed execution of services.

It is also the first project to bundle the efforts of multiple groups from all over the world to tackle difficult problems in information extraction for the Life Science. We believe that such a bundling is the only feasible way to cope with the tremendous challenges in this area; no single group is able to build up the competence and the systems to address merely the most pressing problems [9].

There is an interesting relationship between an approach like ours and current developments in the Mashup-community. For instance, companies like OpenCalais offer a range of services, which perform tagging of texts with entities such as person names, telephone numbers, places etc. Such functions are also frequently available in Mashup-Tools like Yahoo Pipes or Intel's Mash-Maker. However, none of these tools is capable of integrating various tools for the same task and to perform consensus computation on these resources. Clearly, the functionality offered by the BC-VisCon service is unique in itself. It offers a wealth of options to compare and aggregate NER services that is unrivaled by existing tools. We hope that VisCon will become a standard service to be used in text mining competitions and projects building on different (and competing) services, and we also think about adding it as pluggable service to current Mashup-platforms.

5. Discussion

Information extraction has become vital for advancing Life Science research [1]. It is used both in the academic community and in commercial companies. Actually, several large pharmaceutical companies recently built up knowledge management departments to harvest the large amount of information existing, yet not easily accessible, in their corporate knowledge bases (personal communications). As important as mining publications is the extraction of information from patents. Although several companies are active in this area (such as Temis, BioAlma, LingPipe, etc.), no single vendor has the resources to build applications that would cover all customers' needs. On the contrary, commercial tools are regularly outperformed by academic algorithms [10]; However, academic tools are usually specialized to a single problem, have no integration with other tools, and are difficult to deploy and maintain.

The goal of our endeavor is to show the feasibility of building high-end text mining systems for the Life Sciences based on a service-oriented architecture. Such a system is capable of integrating the many freely available methods developed throughout the world, provided they implement certain simple and standardized interfaces. Since the beginning of the BCMS initiative, the platform is being used as a starting ground to several other projects. Examples are services for annotating genome browsers with Medline abstracts (unpublished) and the use of the platform for full-text annotation. This second approach is currently pursued in the BioCreative II.5 challenge, which is run by collaboration between Elsevier/FEBS Letters, the MINT database, and the BioCreative organizers[7]. Apart from these developments, BCMS itself evolves, and new functionality is added to the system in the form of value-added services. BC-VisCon is one such example.

We envision several other applications or extensions to our project. For instance, author curation is a currently discussed possibility to shift the burden of knowledge extraction from professional curators to the authors of publications. In author curation, authors mark respective entities in their text themselves. The annotation is embedded in the file and can be extracted directly by software [13]. Clearly, such an approach is only feasible when authors are supported in their annotation process, which in turn requires flexible and powerful text mining tools. The aforementioned publishers created the FEBS Letters experiment [6], where authors provide annotations for their publications prior to review since January 2008. BC II.5 will now explore the possibility of augmenting this process with the use of the BCMS platform by providing online annotations on the full-text.

Another line of future development in this area is the usage of text mining for the automatic annotation of biological objects in databases. Here, the task is to extract relevant information on a gene, protein, SNP etc. by using text mining [17]. This task requires uttermost accuracy and therefore currently is performed manually by employees of major Life Science databases; however, this approach does not at all scale with the number of objects to be annotated (think of the millions of genes present in all species of the world) and the rapid increase in available articles. A concerted action which gathers the efforts of

[6] See http://alias-i.com/lingpipe/

[7] See http://www.biocreative.org/

different text mining groups and projects all over the world that develop and integrate services such as the ones presented in this paper could potentially offer a more principled and more scalable approach to this problem.

A tool like BC-VisCon also has the power to support the development of future text mining algorithms. For instance, we are currently building in a function with which users may upload their texts to be annotated together with a gold standard annotation. This functionality can readily be used by organizers of text mining challenges to compare participating tools. It would offer the advantage that all evaluation would be performed online, making the complicated and competition-critical shipping of test data unnecessary. Equally well, such a function could be used by companies to perform an online evaluation of the tools of different vendors on their particular types of documents.

Finally, we want to note that our architecture is by no means restricted to the Life Sciences. Actually, BC-VisCon is completely independent of any application domain. Therefore, we also explore the use of our approach to other areas, such as blogs or news mining.

Acknowledgments

The authors would like to especially thank all the groups and researchers providing BC Annotation Servers for their constant effort, in alphabetical order: William A. Baumgartner, Yan Hua Chen, Frédéric Ehrler, Günes Erkan, Barry Haddow, Jörg Hakenberg, Robert Hoffmann, Chun-Nan Hsu, Hsi-Chuan Hung, Lawrence Hunter, Calvin A. Johnson, Sun Kim, Michael Krauthammer, Cheng-Ju Kuo, William W. Lau, ThaiBinh Luong, Michael Matthews, Arzucan Özgür, Conrad Plake, Dragomir R. Radev, José Manuel Rodíguez Carrasco, Patrick Ruch, Rune Sætre, Soo-Yong Shin, Richard Tzong-Han Tsai, Xinglong Wang, Kazuhiro Yoshida, and Byoung-Tak Zhang. FL additionally would like to thank Martin Krallinger whose support and contributions only made the BCMS platform possible.

BioCreative is supported by donations from the ENFIN European Commission FP6 Programme NoE LSHG-CT-2005-518254. JS is supported by a grant from the German Ministry of Education and Research (BMBF). The research group at the Spanish National Cancer Research Centre (CNIO) is funded by the National Institute for Bioinformatics (www.inab.org), a platform of "Genoma España".

References

[1] Agarwal, P. and D.B. Searls, *Literature mining in support of drug discovery.* Brief Bioinform, 2008. **9**(6): p. 479-92.

[2] Augen, J., *Information technology to the rescue!* Nature Biotechnology, 2001. **19**(6).

[3] Bachlehner, D. and K. Fink, *Semantic Web Service Research; Current Challenges and Proximate Achievements.* International Journal of Computer Science and Applications, 2008. **5**(3b): p. 17-140.

[4] Blaschke, C., L. Hirschman, and A. Valencia, *Information Extraction in Molecular Biology.* Briefings in Bioinformatics, 2002. **3**(2): p. 1-12.

[5] Bujnicki, J.M., et al., *Structure prediction meta server.* Bioinformatics, 2001. **17**(8): p. 750-1.

[6] Ceol, A., et al., *Linking entries in protein interaction database to structured text: The FEBS Letters experiment.* FEBS Letters, 2008. **582**(8): p. 1171-7.

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

[8] Ferrucci, D. and A. Lally, *UIMA: an architectural approach to unstructured information processing in the corporate research* Natural Language Engineering, 2004. **10**(3-4): p. 327-348.

[9] Fluck, J., et al., *Information extraction technologies for the life science industry.* Drug Discovery Today: Technologies, 2005. **2**(3): p. 217-224.

[10] Hirschman, L., et al., *Overview of BioCreAtIvE: critical assessment of information extraction for biology.* BMC Bioinformatics, 2005. **6(Suppl 1):S1**.

[11] Krallinger, M., et al., *Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge.* Genome Biol, 2008. **9 Suppl 2**: p. S1.

[12] Leitner, F., et al., *Introducing meta-services for biomedical information extraction.* Genome Biol, 2008. **9 Suppl 2**: p. S6.

[13] Leitner, F. and A. Valencia, *A text-mining perspective on the requirements for electronically annotated abstracts.* FEBS Letters, 2008. **582**(8): p. 1178-81.

[14] Leser, U. and J. Hakenberg, *What Makes a Gene Name? Named Entity Recognition in the Biomedical Literature.* Briefings in Bioinformatics, 2005. **6**(4): p. 357-369.

[15] Mantovaneli Pessoa, R., et al. *Enterprise Interoperability with SOA: a Survey of Service Composition Approaches*. in *Workshop on Enterprise Interoperability*. 2008.

[16] Weiss, S.M., et al., *Text Mining*. 2005, New York: Springer.

[17] Winnenburg, R., et al., *Facts from text: can text mining help to scale-up high-quality manual curation of gene products with ontologies?* Brief Bioinform, 2008. **9**(6): p. 466-78.

[18] Zhou, D. and Y. He, *Extracting interactions between proteins from the literature.* J Biomed Inform, 2008. **41**(2): p. 393-407.