A Cloud-based Approach to Medical NLP

Kyle Chard[1] PhD, Michael Russell[1], Yves A. Lussier[2*] MD,
Eneida A Mendonça[3*] MD, PhD, Jonathan C. Silverstein[4*] MD
[1]Computation Institute, The University of Chicago, IL
[2]Section of Genetic Medicine, Dep. of Medicine, University of Chicago, IL
[3]Dep. of Biostatistics and Medical Informatics, University of Wisconsin-Madison, WI
[4]NorthShore University HealthSystem, Chicago, IL

* Corresponding authors

Abstract
Natural Language Processing (NLP) enables access to deep content embedded in medical texts. To date, NLP has not fulfilled its promise of enabling robust clinical encoding, clinical use, quality improvement, and research. We submit that this is in part due to poor accessibility, scalability, and flexibility of NLP systems. We describe here an approach and system which leverages cloud-based approaches such as virtual machines and Representational State Transfer (REST) to extract, process, synthesize, mine, compare/contrast, explore, and manage medical text data in a flexibly secure and scalable architecture. Available architectures in which our Smntx (pronounced as semantics) system can be deployed include: virtual machines in a HIPAA-protected hospital environment, brought up to run analysis over bulk data and destroyed in a local cloud; a commercial cloud for a large complex multi-institutional trial; and within other architectures such as caGrid, i2b2, or NHIN.

Introduction/Background
There is a wealth of information contained in the text of electronic health records that, if processed, can be used for a range of medical purposes including, but not limited to: clinical decision support, institutional auditing and billing, clinical quality improvement and research. While automated analysis tools exist, general analysis is often a painstaking process due to the quantity and structure of the data coupled with the multi-step processes involved in the use of today's Natural Language Processing (NLP) software. This difficulty extracting meaningful structured data is reflected by end user tools typically consisting of command line interfaces with complex parameters. Our goal in coupling NLP systems to simple end user tools, leveraging cloud approaches, is to bridge this usability gap and make underlying NLP engines available to a wider range of less technical users.

The critical step towards automated analysis of text is obtaining structured information from which subsequent processing can be performed. NLP applications are widely used in many domains as a means of processing unstructured free text. In the medical field, NLP engines not only parse raw text but also encode medical concept mappings so that they can be used by automated tools. Three of the most widely used NLP engines are MetaMap[1], MedLEE[2] (Language Extraction and Encoding) and cTakes[3] (Mayo clinical Text Analysis and Knowledge Extraction System). The current "state of the art" in medical NLP relies on a raw form of processing, in which expert users format (pre-process) and submit unstructured medical records through a command line interface. They then interpret the output results with the aid of proprietary post processing scripts. This is a time consuming and error prone task that requires a great deal of domain and NLP knowledge. Consequently there is a dearth of applications that make direct use of NLP data.

Standardized service interfaces and user tools will help bridge this gap by making underlying NLP engines more usable to a wider range of less technical users. In addition there are several features of current NLP technology that limit their effectiveness in real-world use cases. It is widely acknowledged that many NLP tools are not fast enough for real-time use[4] and that there is much focus on competition for accuracy between NLP engines (e.g. https://www.i2b2.org/NLP/Medication/, the latest in a series of challenges published in JAMIA). One way to mitigate these problems is batch processing and then indexing results; real-time analysis can then be performed on bulk structured data. However, we submit that better approaches may be to leverage cloud approaches to accelerate real-time usage across multiple virtual machines while supporting multiple (potentially parallel) NLP engines, thus providing user-directed complex querying into the texts, supporting user enrichment, and also the potential to improve the mapping quality generated. Standardized, distributed interfaces may also make it possible to collaborate over text analyses with other individual researchers.

Many of the popular NLP tools (MetaMap, MedLEE, cTAKES) have been designed for use in a centralized deployment rather than a scalable, elastic distributed deployment (e.g. Amazon's Elastic Compute Cloud - EC2). This is especially evident as they do not offer an integrated service interface (MetaMap has an additional Java API) and there is a lack of client side tooling for interacting with processed narratives. The Smntx (Semantic Mining of Natural TeXt) architecture presented in this paper takes the opposite approach by supporting different underlying NLP engines and providing a service based architecture to support distribution and use via different tools including a client side web interface to facilitate intuitive non-expert use.

One project that has taken a service oriented approach to medical NLP is the Cancer Text Information Extraction System (caTIES)[5]. caTIES is designed to support collaborative tissue banking and text mining. Collaborative research is a major motivation for this work and as such a great deal of focus has been put towards sharing data within privacy regulations. A federated data model is used to access diverse research data sources and a security model is provided for multi centre research. The general NLP workflow pares free text, maps phrases to a limited set of concepts and extracts a result hierarchy to XML. The caTIES workflow is constructed using a combination of GATE[6] (General Architecture for Text Engineering) and custom components, MMTx (MetaMap Transfer) is used to map concepts to fragments of free text. The architecture is composed of a group of WSRF (Web Services Resource Framework) Grid services wrapping both functionality and data sets. A custom Apache Lucene index provides similar search capabilities to the architecture we will describe here, however without complex functionality like faceting. A feature rich Java application provides a graphical user interface (GUI) to individual users, perspectives can be configured to match user requirements. This GUI includes text highlighting and a novel graph based query composition tool to create complex temporal queries.

In this paper we first outline an example set of user scenarios we are addressing to ensure clarity in purpose for a cloud-based medical NLP architecture. Then the majority of the paper outlines a scalable architecture developed to support these scenarios which is extensible to a wide variety of related uses.

In short, we describe how a scalable architecture, such as the architecture we've engineered, can address the broad problems currently limiting NLP usage via: 1) deployment of a distributed service-based architecture in a scalable cloud environment with a well-defined service interface to medical NLP engines (thereby facilitating usage with a variety of end user tools); 2) integrating an enterprise search architecture to facilitate efficient, complex mining and exploration of data; 3) exposing these functions through an easy to use web interface that supports intuitive single and multi document navigation and user enrichment. We've titled our engineered architecture and applications: Smntx (pronounced as semantics).

User Scenarios

In the medical domain there are a large number of potential users of NLP processing, visualization and data mining. For example, clinicians and researchers can more quickly analyze documents and explore relationships over a much larger pool of documents. In addition entire medical institutions can benefit through automated reporting, accounting and auditing. The remainder of this section presents four specific scenarios in which there are immediate, and largely untapped, opportunities for value of NLP over clinical texts.

a) **Medical Coder:** Medical coders process medical claims on behalf of physicians and hospitals to receive reimbursement from insurance companies for services and facilities provided to patients. Coders typically follow a manual process of examining individual medical documents to determine the services provided. Using Smntx this process can be optimized, allowing coders to compose queries to identify only pertinent data (e.g only International Classification of Diseases (ICD) concept hits) and also drill down through web-based data visualization panels to view the specific context for the concept to ensure accuracy.

b) **Clinicians:** Individual physicians must analyze a wide range of documents (e.g entire patient history) when determining diagnoses and planning therapy. This process could be also be optimized by enabling more efficient searching over a range of different medical documents and also visualizing concept queries as an overlay on top of individual documents thus increasing analysis speed or scope of documents reviewed in an allotted time.

c) **Cohort Discovery:** Appropriate patient identification and selection for clinical trials is a crucial process which dramatically influences the success of clinical research. In many cases, patient selection is a complex and time consuming process in which researchers manually review patient information to determine trial eligibility. Multiple factors, spread over an entire patient's history, can potentially be considered in this selection. Smntx can be used to explore and query entire datasets to determine which patients match general inclusion and exclusion criteria and to see the matching phrase in context for verification.

d) **Retrospective chart review:** Retrospective chart review is used both as a quality improvement aid and as a method of research. Smntx can be used to classify terms in a large number of medical record documents which can then be filtered (real-time) by user defined queries. Physicians or researchers can explore the collection of analyzed documents following specific hits as links into documents or querying for particular concepts. For example a physician may want to view a list of all patients prescribed a particular medication or they might want to look at how specific medications have effected patients with different diseases.

Smntx Architecture

The Smntx architecture is composed of three major components, these are: the NLP service wrapper, distributed persistence and search backend repositories, and the user interface. The high level architecture of Smntx is shown in Figure 1. This figure illustrates the versatility of the service-based architecture as three diverse interfaces are shown to be (simultaneously) interacting with the same backend services. In this diagram a clinician is shown using the web interface, a researcher is using a high level programming language, and an entire institution is utilizing automated scientific workflows to process bulk data. These components are described in the subsequent sections of this article.

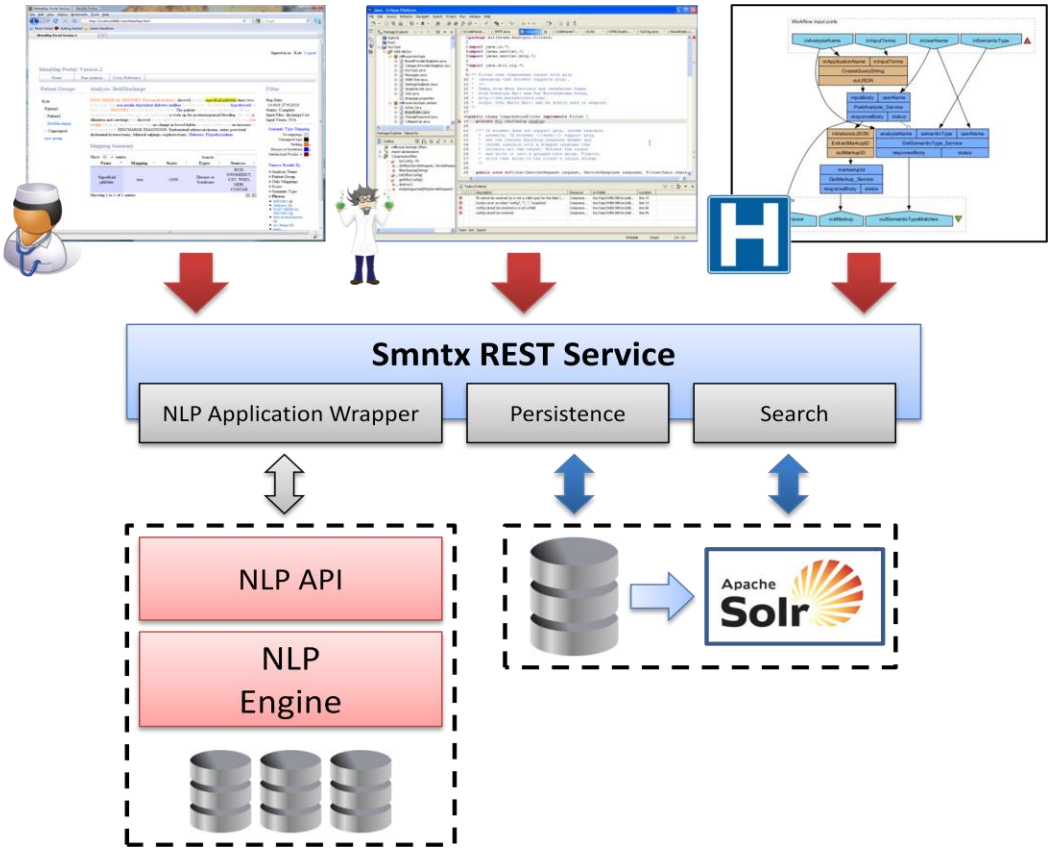


Figure 1: Smntx high level architecture

a) NLP Engine : At the core of the Smntx architecture is the NLP engine used to parse unstructured medical text and map terms against a medical metathesaurus. Smntx is designed to "wrap" different NLP engines through a simple plug-in architecture. Currently, Smntx includes a MetaMap NLP wrapper. MetaMap is a generic NLP platform developed by the National Library of Medicine which combines a number of processes and algorithms to map concepts from the Unified Medical Language System (UMLS) Metathesaurus to medical text. The main MetaMap application is based on two services: tagger and (optionally) word sense disambiguation (WSD)[7]. The tagger is responsible for tokenization, part of speach tagging, and concept mapping. WSD attempts to overcome ambiguity byfavoring concept mappings which are consistent with surrounding text. Together, these two services produce a set of concept mappings for terms in the text.

MetaMap is a prolog-based application and is desgined to be used through a command line interface, it also includes a Java API layer (using prolgbeans) which can be used to remotely interact with the running service. Smntx makes use of this API to remotely process documents and retrieve results. Other non web-enabled NLP engines would require a similar wrapping process. When analysis is started through Smntx a resource is created to represent that NLP analysis, each analysis is queued and sent to a MetaMap engine for processing when there is availability. Results and provenance data is stored in persistent databases; results are also indexed in Apache Solr. Service interfaces are provided to start, manage, and retrieve running instances. Smntx also maintains provenance metadata regarding options and flags used when running an analysis and also the tool used (in this case MetaMap), such that every result is reproducible.

One major advatange of MetaMap is the level of configurability across multiple dimensions, for example data options, data models, output options, and processing options can all be customized. It is important to note however, that many of these options reduce the data set produced, which is contrary to the Smntx model. In the Smntx model this reduction should occur through queries or filters placed on the fully processed results, as this does not diminsh the data set and makes subsequent querying more efficient.

b) Persistence and Indexing: The Smntx model attempts to reduce repeated processing through a single NLP stage and the application of real-time filters to access processed data. To achieve this persistence and still provide high performance filtering Smntx separates this process using two tools: coded results and provenance data is stored in a durable relational database, all results and raw text are also indexed to optimize performance and support complex queries, full text search, and faceted search. In the current implementation, the relational database used is Apache Derby, a lightweight java database capable of operating in both embedded and network mode. Indexing and full text search is provided by Apache Solr; an enterprise search platform based on the Apache Lucene search library. Both relational databases and Solr are highly scalable due to their distributed architectures, Solr also supports both distributed search and index replication.

c) REST Service Architecture: Representational State Transfer[8] (REST) is a client-server service architecture based around the transfer of representations (documents) of resources. A RESTful web service is a web service implemented using HTTP and following the princples of the REST architecture. Briefly, a REST service represents a collection of resources identified by a URI which can be interacted with using a set of defined HTTP operations (GET, POST, PUT, DELETE).

There are several advantages obtained by using a REST model. First, it provides a simple abstraction layer on top of a complex backend architecture (NLP, enterprise search, persistent storage). Moreover, this layer is easily consumable in different environments and can be "wrapped" with other more complex functionality depending on the environment in which the services are deployed, for example utilizing different authentication and authorization mechanisms (HTTP basic, GSI security using x509 certificates) or secure communication channels (SSL). Secondly, the Smntx REST service is agnostic to architecture, that is, Smntx can function as a "plug-in" in different host architectures. For example, Smntx could be trivially deployed in SOA based architectures like cancer Biomedical Informatics Grid (caBIG)[9], ,informatics for Integrating Biology and the Bedside (i2b2) Hive[10], and the Nationwide Health Information Network (NHIN Direct). These deployment opportunities are discussed in detail in the following sections. Finally, the service based approach used in Smntx supports distribution in Cloud scenarios allowing multiple Smntx instances to be deployed over a distributed network of hosts, each sharing the same backend database for context while potentially offering access to heterogeneous NLP engines.

The Smntx REST service represents four different resource types (user, NLP analysis, HTML markup, and search). Data passed through the API is generally JSON based, and in most cases the service accepts either a JSON object, query parameters or form encoded parameters. Responses are returned as JSON objects or HTML pages. The Smntx REST mappings are shown in Table 1. The *user* resource abstracts user information and provides an interface to register users, authenticate users, store user information, upload and download files, create groups, and group analyses. The *NLP analysis* resource provides the ability to start NLP processing and, update, delete and retrieve previously run analyses. The *markup* resource stores HTML markup of the raw text document to support client side highlighting and interactive HTML filtering, this resource is typically only used by the web interface. Finally the *search* resource abstracts interaction with Solr (e.g full text search, general search and faceted search) and also provides an interface to save and retrieve common queries.

Table 1. Summary of the Smntx REST service mappings

Resource Type	URL Mapping	Type Returned	REST Operations
User	/users/{username}	JSON	GET, PUT, DELETE
User	/users/{username}/groups/	JSON	GET, POST (All groups)
User	/users/{username}/groups/{gpname}	JSON	GET, PUT, DELETE
Analysis	/analysis/{NLPTool}	JSON	POST (start analysis)
Analysis	/analysis /{analysisId}	JSON	GET, PUT, DELETE
Analysis	/analysis /{analysisId}/files	JSON	GET
Analysis	/analysis /{analysisId}/files/{fname}	JSON	GET
Markup	/analysis /{analysisId}	HTML	GET
Markup	/analysis /{analysisId}?{phraseId}	HTML	GET
Search	/search?query_params	JSON	GET, POST
Search	/search/saved/{searchid}	JSON	GET, POST, PUT, DELETE

d) Deployment - Smntx in the Cloud

Cloud computing is a scalable elastic computing model in which virtualized resources are provisioned on demand by consumers. Software as a Service (SaaS) forms the top layer of the Cloud stack, offering specialized well defined application services to consumers. Smntx follows a SaaS approach through a modular service based architecture which allows consumer access without requiring knowledge of the underlying infrastructure or technology. One of the major advantages of this approach is that the Smntx application can scale on demand as additional virtual machines are created and deployed. Moreover, the backend services can be optimized through parallelization (for example the NLP engine) and "sharding" (distributing) the data repositories and indexes. In addition, small scale instances of Smntx can be trivially instantiated, deployed and destroyed to rapidly process data sets for a fixed period of time. The Smntx SaaS architecture has been verified through a Cloud deployment on local resources and a production deployment to the HIPAA compliant Cloud at the University of Chicago in the Computation Institute's Initiative in Biomedical Informatics.

e) Interoperability with other platforms

Web services, by definition, support interoperable machine-to-machine interaction through self-describing standardized interfaces. As previously stated, this is a considerable advantage of the Smntx REST model as it allows "plug-in" usage in a number of systems such as caBIG, i2b2 Hive, and NHIN Direct. Deployment in each of these systems potentially exposes Smntx to a large number of end users. This section provides a brief overview of each of these systems, highlighting the ease in which Smntx can be used in each environment.

caBIG is a distributed collaborative information network designed to accelerate cancer research. The caBIG infrastructure is built upon a distributed network of machines which facilitate sharing of tools and data. caBIG follows a strong service oriented architecture in which all tools and data repositories are exposed through web services. Due to the potentially sensitive nature of the data in caBIG, a comprehensive security architecture is used. The Taverna workflow engine is used to orchestrate diverse services into a coherent workflow. Due to the standardized nature of the caBIG infrastructure Smntx can be easily deployed such that all members of the caBIG community can interact with both the web interface and REST services. Deployment in the caBIG environment requires layering of secure communication channels and authentication mechanisms which are natively supported by Smntx.

i2b2 Hive is a collection of interoperable loosely coupled web services (or cells) that can be aggregated to perform particular tasks. User interaction is supported at the individual cell level or through orchestration of many cells using workflows The modular nature of the Hive allows additional services to be added without requiring low level integration between services. Smntx can therefore be deployed as a standalone cell in Hive allowing users access through the same mechanisms that they currently use. Moreover, Hive users would then be able to utilize Smntx functionality when developing and running scientific workflows.

NHIN is a public-private network that supports the secure exchange of healthcare information between providers, consumers and other parties. As a part of this network, consumers are able to retrieve their healthcare data in different locations so that it can be used in clinical decision support. As part of NHIN, participants are developing standards based data transfer mechanism as well as new capabilities for managing and controlling personal health records. The Smntx model could be valuable in such an environment both as an end user tool for consumers as well as assisting decision support efficiency with the increasing volume of data available to clinicians.

f) Privacy and Security: Given the strict regulations present in the medical domain, privacy and security are of the utmost importance in Smntx. At the lowest level, for identified patient information, it is assumed that both the NLP engine and the Smntx services are hosted in a HIPAA compliant environment. In a less secure environment only anonymomized data should be used. All interactions with Smntx must be performed with an authenticated user and users must register before being able to use any of the services, but the authentication model can be quite flexible given the light-weight architecture. At this time, data is only accessible to the user that created it but this could be expanded. We aim to explore data sharing policies to increase data set size and reduce computational overhead. In addition to Smntx authentication, the UMLS data store requires that all users have UMLS Terminology Services (UTS) licenses. To support this Smntx is an authorized content distributor, which means that end users can use Smntx with their UTS credentials. Smntx is the first user of the UTS REST authentication service provided by National Library of Medicine to authenticate the users' credentials before allowing them access to UMLS data. Thus this third-party UMLS license verification provided via REST interface ensures underlying libraries are licensed to the user with which we interact. Similar approaches could be used for other necessary components in an analysis (e.g. NLP engine).

g) User Interactions: Smntx was primarily designed for use in three important interaction modes: the web interface, scientific workflows, or programmatically.

1. Web Interface: The main focus of Smntx is the web interface as it provides the most complete user experience. After authenticating with the service, users are able to start new processing, navigate previously processed documents, and search over previously processed documents. At all times a tree-like view of all documents analyzed is presented and documents are grouped (for example by patient) to simplify navigation. When starting new processing the user is guided through a web form to define properties of the analysis, for example setting the group, the analysis file or text, and NLP specific properties. The resulting task can then be run either synchronously or asynchronously.

   A customizable individual document pane is used to view HTML markup analysis of a single document. This pane is designed to make user analysis more efficient by highlighting selected terms (based on semantic type). Figure 2. shows an example of a processed discharge summary. A customizable highlighting list is shown on the right. In this example three common groupings are applied to highlight disorders, procedures and medications in the text. Clicking on any specific phrase in the text can be used to show a tabulated view of a specific term and it's mappings.

   A more traditional data grid provides a structured view of multiple documents simultaneously. Multiple document views are often used when looking over patient history or as the result of a query across data sets. The multi document data grid view is shown in Figure 3. The grid itself is completely customizable in terms of what fields are shown, result ordering, and number of results displayed per page. A client side search tool is also provided to search through all the results displayed in the grid. Individual results can be "expanded" to provide context (surrounding sentences) in which the phrase occurs. This view also allows expert users to improve or enrich the generated mappings. For example, individual values can be altered and entire mappings can be deleted.

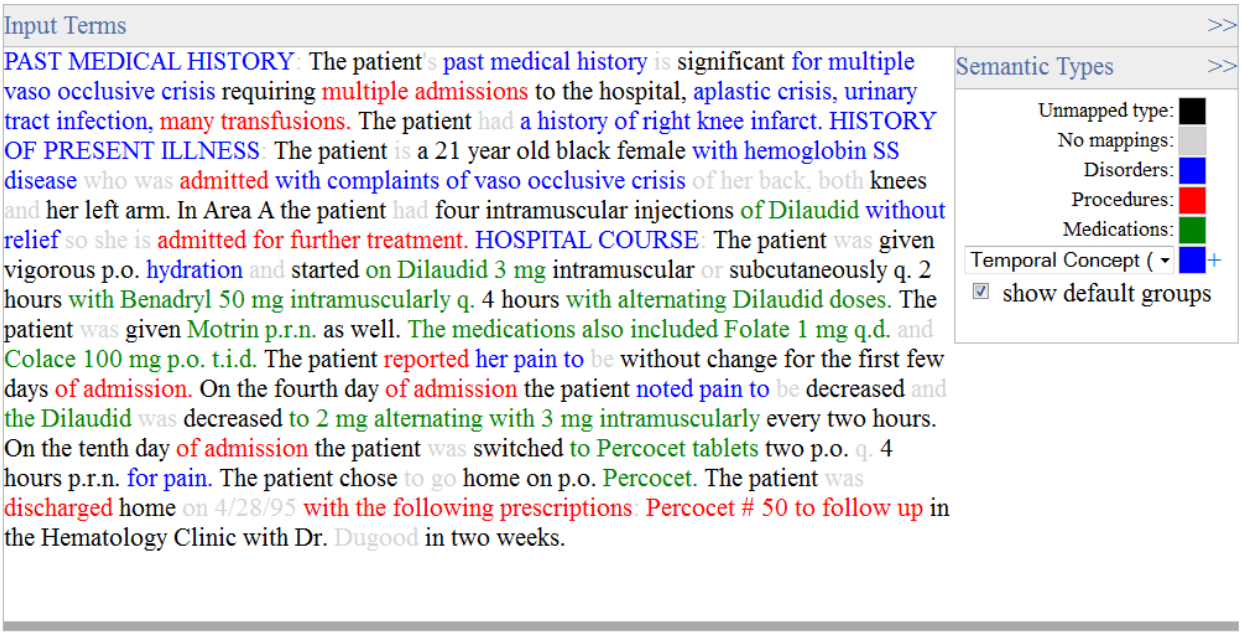


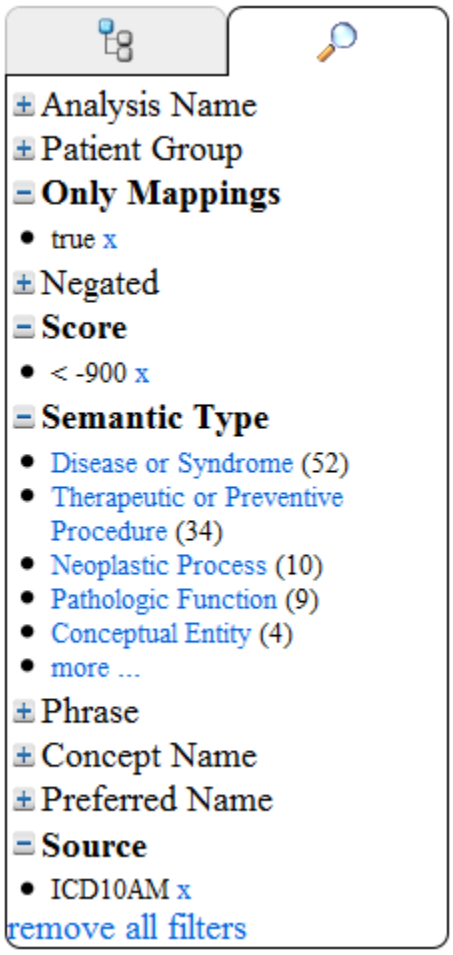
Figure 2: Smntx document highlighting

Show 10 entries Search:

Analysis Name	Patient Group	Negated	Preferred Name	Score	Semantic Types
discharge1.txt	Patient9	false	Actual Depression	-1000	Finding
discharge1.txt	Patient99	false	Actual Depression	-1000	Finding
Patient was followed by Psych for depression and treated with Celexa.					
RemoteDischarge	Ungrouped	false	Actual Depression	-1000	Finding
discharge2.txt	Patient9	false	Actual Pain	-1000	Finding
discharge2.txt	Patient99	false	Actual Pain	-1000	Finding
discharge1.txt	Patient9	false	Actual Physical Weakness	-1000	Finding
discharge1.txt	Patient99	false	Actual Physical Weakness	-1000	Finding
RemoteDischarge	Ungrouped	false	Actual Physical Weakness	-1000	Finding
Mamo	Patient1	false	Antimicrobial susceptibility	-1000	Finding
discharge1.txt	Patient9	false	Complaint	-916	Finding

Showing 1 to 10 of 153 entries

Figure 3: Smntx data grid view



Fgure 4: Smntx faceted search

Faceted search allows users to view and filter results over a defined set of categories, as is common in online marketplaces such as Amazon. The faceted search panel in Smntx is shown in Figure 4. Each of the indexed categories can be expanded to show a list of the most common results and the number of occurrences. This both summarizes that data as well as provides the ability to perform real-time filtering by selecting on a single value. By selecting an option, a filter is applied and the number of occurrences displayed for the other categories is updated to reflect the current filters.

In addition to the views presented in this section, Smntx includes two specific query views. Firstly, an advanced search page allows users to customize and run complex boolean queries. This view also includes the ability to save and reload commonly used queries. Secondly, a traditional free text search is also provided (similar to online search engines like Google). In this view the search is performed on the un-mapped input text and results are grouped by individual document, therefore providing a means of quickly finding a specific document without needing to navigate through an entire data set.

2. Workflows: Like many scientific domains, medical analysis is increasingly data and process driven. This type of analysis is well modeled by scientific workflows that orchestrate a sequence of operations in a web-scale manner. Consider the example of hospital wide auditing; A workflow can be defined to automatically process medical text as it is generated; The workflow can include operations to look for key concepts (specific diseases) or combinations of events and then report back through notifications or custom summaries.

   To test this approach we have constructed both interactive and batch workflows in Taverna[11]. Using Taverna, workflows can be simply created, modified and executed through the Workbench GUI. The task of the developer is reduced to discovering and adding REST Activity plug-ins for each particular operation they want to execute. Workflows have several advantages over other methods: 1) complex workflows can be constructed by less technical users as operations can be discovered and connected using a simple graph representation; 2) workflows can be shared amongst peers to both extend and validate research; 3) workflows can be trivially run in parallel; 4) data sets and services used can be easily altered without changing the workflow itself. An example Taverna workflow using the Smntx REST API is presented in Figure 5. This workflow processes entire patient records looking for specific diseases and treatments.

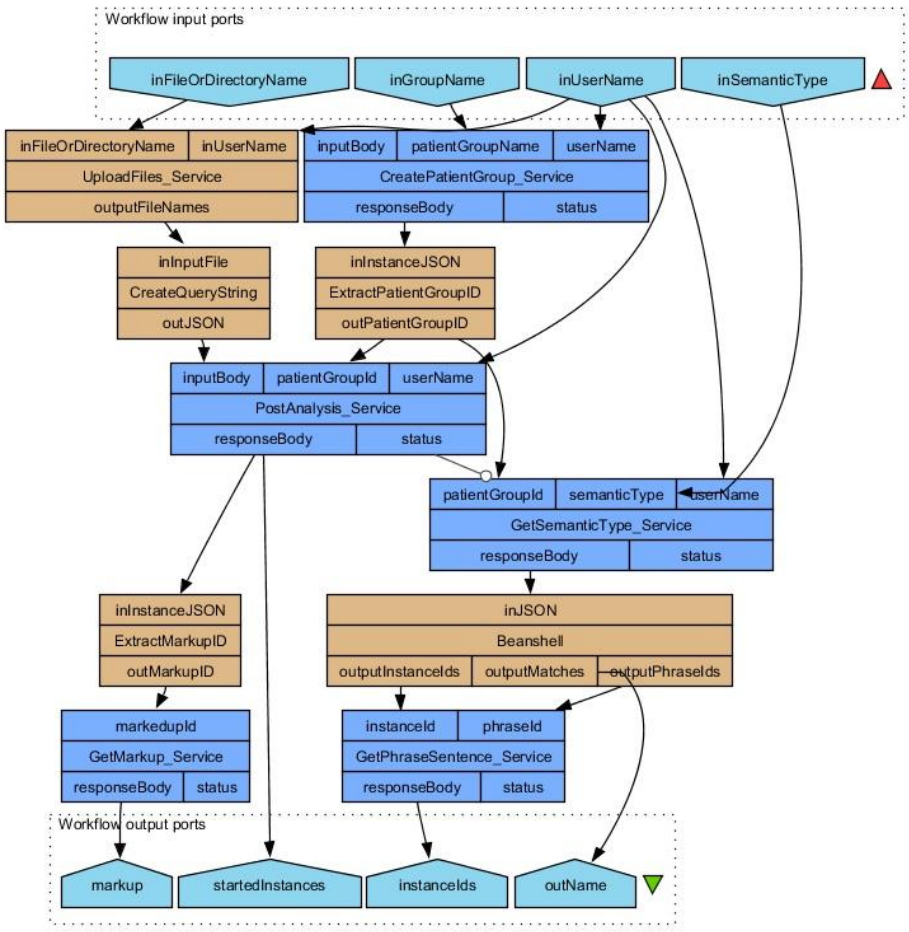


Figure 5: Example scientific workflow using Taverna and the Smntx REST API

3. Programmatically: A greater degree of flexibility can be obtained by using high level programming languages or scripts to perform analysis. The Smntx REST API gives developers complete control of how the analysis is conducted and how the results are retrieved, this is especially valuable when pre and post processing stages are required. One of the major advantages of the REST API is that it is lightweight and can be accessed through simple client side web based libraries/packages available in all high level programming languages without the requirement for customized client APIs.

Future Work

The continuing work on Smntx focuses on three broad areas: performance, extension and dissemination. Detailed performance assessment looking at service overhead, search cost, and scalability needs to be done in a deployed environment. We anticipate doing this internally at NorthShore University HealthSystem.

In terms of extension there are various areas we plan to explore. One of our first goals is to extend the current deployment by adding additional NLP wrappers. In this way the quality of mappings may be improved by leveraging analysis of different engines. Secondly, there is potential to provide feedback into the NLP process based on user input (deleting, editing). As a first step MetaMap supports simple user extension through the definition of synonyms, this capability could be supported through Smntx. When multiple NLP engines are available there is potential to correlate results across analysis. This information can be used as a form of feedback to selectively favor mappings from particular engines. Native feedback integration is a much more complex task and requires the support at the NLP engine level.

Dissemination is, at present, the major goal of this work, we are currently exploring avenues in which Smntx can be deployed to clinical and research communities. A large part of this work will focus on user studies to tailor the

current implementation to specific communities. We aim to develop evaluation criteria by which we can evaluate the impact of Smntx and to improve its results. Finally we have also looked at generalizing the approach to other scientific domains, allowing users to analyze and index text based data over diverse scientific data sets.

Conclusion

Medical NLP provides the basis for automated text analysis for many medical processes, ranging from clinical decision support through to large scale research projects. While many NLP engines exist, most require a high degree of domain expertise and technical knowledge to be used effectively. Moreover, little emphasis has been placed on scalability and end user tools to provide real-time access, visualization and mining of resulting data sets. We believe that a cloud-based approach using virtual machines and REST services will create a flexible scalable architecture that is agnostic to the NLP engine used. Finally, this standardized approach facilitates flexible deployment scenarios and simplified creation of non-expert tools that abstract he complexities of NLP technology, it is our hope that this will lead to greater adoption in real world scenarios.

The Smntx architecture we describe takes a user-oriented approach to medical NLP applications by focusing on usability and scalability in a range of application scenarios. Smntx is a REST service-based architecture in which NLP applications are exposed through a generic interface that can be trivially consumed by different end user tools. The resulting NLP analysis is stored and indexed such that users can explore results using complex faceted queries and improve results by updating individual mappings. The web interface is designed to expose the core functionality in an intuitive manner, supporting both individual file analysis with customizable text highlighting and a multi-document data grid view.

Acknowledgements

This project is supported in part by the NIH Clinical and Translational Science Awards (5UL1RR024999, 5U54CA121852)

References

[1] Aronson, AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the metamap program. Proc of the AIMA Symp. 2001. 17-21.

[2] Friedman, C. Hripcsak, G. DuMouchel, W. Johnson, SB. and Clayton PD. Natural language processing in an operational clinical information system. Journal of Natural Language Engineering 1995;1(1): 83-108.

[3] Savova, GK, et al. Mayo clinical Text Analysis and Knowledge Extraction. J Am Med Inform Assoc 2010; 17: 507-513.

[4] Aronson, AR, and Lang FM. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc 2010; 17: 229-236.

[5] Crowley, RS. Castine, M. Mitchell, KJ. Chavan, G. McSherry, T. and Feldman, M. caTIES—A Grid Based System for Coding and Retrieval of Surgical Pathology Reports and Tissue Specimens In Support Of Translational Research. J Am Med Inform Assoc 2010;17(3): 253-264.

[6] Cunningham, H. Maynard, D. Bontcheva, K.. and Tablan, V. GATE: an Architecture for Development of Robust HLT, Proc of ACL-02. 2002. 168-175

[7] Humphrey, SM, Rogers, WJ. Demner-Fushman, D. and Rindflesch, TC. Word sense disambiguation by selecting the best semantic type based on journal descriptor indexing: preliminary experiment. J Am Soc Inf Sci Technol 2006;57(1):96-113.

[8] Fielding, RT. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.

[9] Saltz, J, et al. caGrid: Design and Implementation of the Core Architecture of the Cancer Biomedical Informatics Grid." Bioinformatics 2006;22(15): 1910-1916 .

[10] Mendis, M, et al. Integration of Hive and Cell Software in the i2b2 Architecture. Proc of AMIA Sym.p 2007. 1048.

[11] Oinn, T, et al. Taverna/MyGrid: aligning a workflow system with the life sciences community. In Workflows for E-science: Scientific Workflows for Grids, by E. Deelman, D. B. Gannon, and M. Shields I. J. Taylor, 300-319. Springer-Verlag, 2007.