Scholars' Mine

Doctoral Dissertations

Student Theses and Dissertations

Fall 2017

Cognition-based approaches for high-precision text mining

George John Shannon

Follow this and additional works at: https://scholarsmine.mst.edu/doctoral_dissertations

Part of the Artificial Intelligence and Robotics Commons, Linguistics Commons, and the Operations Research, Systems Engineering and Industrial Engineering Commons

Department: Engineering Management and Systems Engineering

Recommended Citation

Shannon, George John, "Cognition-based approaches for high-precision text mining" (2017). *Doctoral Dissertations*. 2630.
https://scholarsmine.mst.edu/doctoral_dissertations/2630

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

COGNITION-BASED APPROACHES FOR HIGH-PRECISION TEXT MINING

by

GEORGE JOHN SHANNON

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

SYSTEMS ENGINEERING

2017

Approved by
Steven Corns, Advisor
Donald Wunsch, Co-Advisor
Suzanna Long
Cihan Dagli
Henry Wiebe
Ruwen Qin

ABSTRACT

This research improves the precision of information extraction from free-form text via the use of cognitive-based approaches to natural language processing (NLP). Cognitive-based approaches are an important, and relatively new, area of research in NLP and search, as well as linguistics. Cognitive approaches enable significant improvements in both the breadth and depth of knowledge extracted from text. This research has made contributions in the areas of a cognitive approach to automated concept recognition in.

Cognitive approaches to search, also called concept-based search, have been shown to improve search precision. Given the tremendous amount of electronic text generated in our digital and connected world, cognitive approaches enable substantial opportunities in knowledge discovery. The generation and storage of electronic text is ubiquitous, hence opportunities for improved knowledge discovery span virtually all knowledge domains.

While cognition-based search offers superior approaches, challenges exist due to the need to mimic, even in the most rudimentary way, the extraordinary powers of human cognition. This research addresses these challenges in the key area of a cognition-based approach to automated concept recognition. In addition it resulted in a semantic processing system framework for use in applications in any knowledge domain.

Confabulation theory was applied to the problem of automated concept recognition. This is a relatively new theory of cognition using a non-Bayesian measure, called cogency, for predicting the results of human cognition. An innovative distance measure derived from cogent confabulation and called inverse cogency, to rank order candidate concepts during the recognition process. When used with a multilayer perceptron, it improved the precision of concept recognition by 5% over published benchmarks. Additional precision improvements are anticipated.

These research steps build a foundation for cognition-based, high-precision text mining. Long-term it is anticipated that this foundation enables a cognitive-based approach to automated ontology learning. Such automated ontology learning will mimic human language cognition, and will, in turn, enable the practical use of cognitive-based approaches in virtually any knowledge domain.

ACKNOWLEDGMENTS

This research would have been impossible without the patience, experience, knowledge, and support of my advisors – Dr. Steven Corns and Dr. Donald Wunsch. To them I wish to express my sincere gratitude. I am also extraordinarily lucky to have been supported as a Chancellor’s Fellow, and by the EMSE department.

I would like to thank Dr. Gayla Olbricht for her consultations on the statistical methods for testing the cognitive relevance measure.

In addition, I wish to thank James Levett, MD, cardiothoracic surgeon and Chief Medical Officer for the Physicians Clinic of Iowa, for his support and innovative ideas.

I am also indebted to Dr. John Barnden and Dr. Andrew Gargett of the University of Birmingham, Department of Computer Science, for their valuable insights that helped guide research priorities.

And I also wish to thank my father, Bill Shannon, for his constant encouragement.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF ILLUSTRATIONS

LIST OF TABLES

Page

1. INTRODUCTION

1.1. ORGANIZATION

The papers included in this dissertation are as follows:

1. *Discovering objective functions for tagging medical text concepts,* conference paper (Appendix A).
2. *Inverse ontology cogency*, submitted to Neural Networks journal (Appendix B), currently under review.
3. *Cognitive relevance*, draft of journal paper for submission to the IEEE Transactions on Biomedical Engineering (Appendix C). Status: pending final review by authors.

Other relevant documents include:

1. *Cognitive Search Test Plan* (Appendix D) – this document outlines the details of testing to be accomplished, with a physician, over the next month. This testing will finish the work necessary to submit the third paper listed above, "Cognitive relevance."
2. *Biography* for James Levett, M.D., CMO of Physicians' Clinic of Iowa (Appendix E). Dr. Levett has agreed to perform the final testing for the cognitive relevance measure, the third paper listed above.
3. *Architecture for Semantic Processing System (SPS)* (Appendix F) – this document contains the architecture definition, analysis, and SysML model developed as the guiding framework for the research discussed in this dissertation. It was developed very early in the research and formed the basis to successfully using a systems perspective to planning and solving problems in natural language processing.

1.2. OBJECTIVES

This research has developed cognitive-based approaches that improve precision of concept recognition. This is required for high-precision extraction of knowledge from text by providing a cognition-based approach to natural language processing (NLP) and information extraction. This includes a completely natural language interface that enables a highly intuitive approach to search.

Furthermore, this research included use of a systems engineering approach. The systems approach identified key algorithmic and technology enablers necessary for high-precision search. The system architecture for cognitive-based approaches, called the Semantic Processing System, or SPS, is provided in Appendix F.

A key finding from the system architecture was the identification of automated ontology learning as the primary enabler for the use of cognitive-based approaches across knowledge domains. The Semantic Processing System framework provided the tool for this assessment. Automated ontology learning, in combination with cognitive approaches to NLP, enables disruptive search and natural language processing technologies that deliver significant improvements in search precision and ease-of-use. And furthermore, this reduces the expense of these approaches such that they are cost-effective across knowledge domains.

1.3. COGNITION-BASED SEARCH AND ONTOLOGY

The objective of this research is high-precision text mining via the use of cognition-based methods. The systems perspective identified an architecture that integrates functions and components important to successfully realize these approaches in real-world products.

As a result of the systems approach to research, confabulation theory [1-4] was identified as the theoretical basis for developing an integrated architecture that delivers high levels of precision. Of key distinction is the use of conceptual and contextual information, extracted from ontologies, to improve the precision of search. The term “cognition-based search” refers to the method of search employing both conceptual and contextual information. An example of the use of confabulation for contextual choices and behavior is provided in [5].

Concept-based and context-based search approaches have been shown to improve precision [6]. These relate as follows:

- Concept refers to an entity that represents a mental notion, stored in the human brain by a neural code, i.e., a collection of neurons.
- Context refers to the cognitive frame of reference, that is, the relations between two or more concepts. Each relation is stored in the brain as one or more neural paths between the neural codes for the related concepts. These neural paths are energized during the cognitive process, which may in turn energize other related paths that, in turn, energize other related concepts, ad infinitum.

This dissertation combines both concept and context-based search into a single notion entitled "cognition-based search." Mental notions stored in the brain, i.e., concept, are represented by a group of neurons. Relationships between concepts, i.e., context, are stored in the cerebral cortex as both the axonal/dendritic links between concepts. These links include a relationship type, where a relationship type is itself a concept.

This dissertation proposes a relationship between confabulation theory, which is a theory of human cognition, and the ontology. Confabulation theory purports that concepts and relations, in combination, form the knowledge base stored in the human cerebral cortex [1-4]. The ontology, which originated with the discipline of philosophy and documents a domain of knowledge [7], likewise consists of concepts and relations between concepts. When combining these two perspectives the ontology can be viewed as an emergent property of confabulation, where the ontology is the human-readable, physical artifact representing the knowledge base stored in the cerebral cortex. Moreover, confabulation theory postulates a process for human cognition that can be reduced to a measure used for the objective optimization of the cognitive process. The potential also exists for mimicking human cognition via the development of an artificial neural architecture that executes the human language cognitive process. For these reasons confabulation was chosen for this research as the cognitive theory to apply to the problems of high-precision search and automated concept recognition.

1.4. RESEARCH CHRONOLOGY

The sequence of research discussed in this dissertation is provided in Figure 1.1, and is provided to link and logically develop multiple topics discussed in this dissertation.

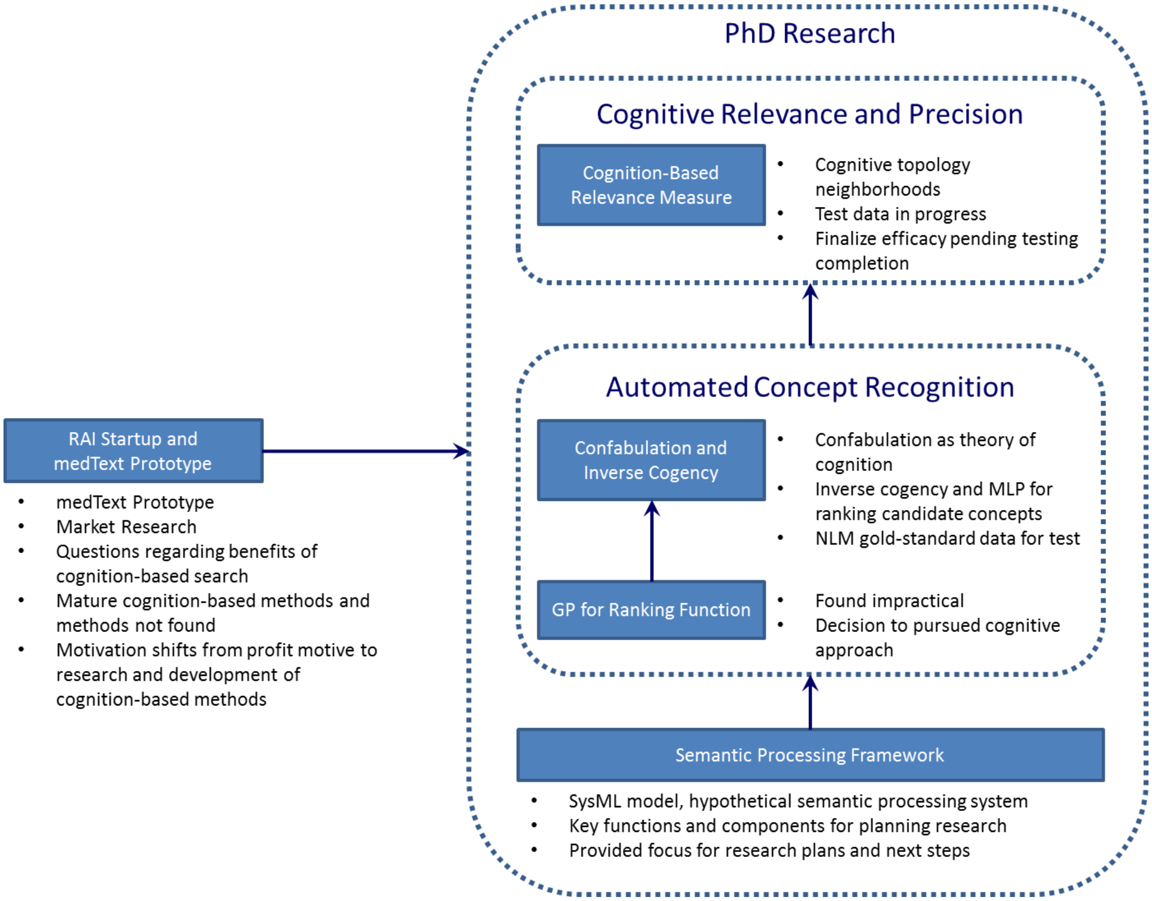

PhD Research
Cognitive Relevance and Precision
Cognition-Based Relevance Measure
• Cognitive topology neighborhoods
• Test data in progress
• Finalize efficacy pending testing completion
Automated Concept Recognition
Confabulation and Inverse Cogency
• Confabulation as theory of cognition
• Inverse cogency and MLP for ranking candidate concepts
• NLM gold-standard data for test
GP for Ranking Function
• Found impractical
• Decision to pursued cognitive approach
Semantic Processing Framework
• SysML model, hypothetical semantic processing system
• Key functions and components for planning research
• Provided focus for research plans and next steps
RAI Startup and medText Prototype
• medText Prototype
• Market Research
• Questions regarding benefits of cognition-based search
• Mature cognition-based methods and methods not found
• Motivation shifts from profit motive to research and development of cognition-based methods

Figure 1.1: Research Chronology: research task sequence, linked by the common goal of developing a cognition-based approach for high-precision search.

Prior to beginning PhD studies at MST, the author led the startup of a small technology company, Raphael Analytics. The targeted product market niche required the ability to provide high-precision search using complex query criteria.

Raphael Analytics, Inc. was a small technology startup that pursued a medical text search product to support clinical processes. This included the development of a prototype of a search tool, medText, which retrieves clinical information from medical text. During this development a number of questions were identified about search, most notably regarding the potential benefits of a cognitive-based approach and the lack of available mature, proven algorithms and software libraries that used cognitive approaches.

During this startup, questions arose regarding the choice of search methods and technologies that deliver high-precision search. These questions were motivated by the need for a new product with profit-generating economic and engineering benefits. But, when it became apparent that the required algorithms and tools did not yet exist that met the need for high precision, this motivation evolved into pursuit of a PhD with the purpose of research in cognitive-based methods capable of high-precision search. While the research discussed in this dissertation focuses on technology innovation, this motivation remains linked to the potential for commercial products capable of wealth creation and how this research can enable these products.

1.5. AUTOMATED ONTOLOGTY LEARNING, COST-BENEFIT THRESHOLD FOR FEASIBILITY

During the development of the SPS architecture in this research, it became apparent that medicine is, arguably, the only domain where large, complicated ontologies are readily available. Ontologies developed outside of healthcare are limited to smaller, grossly simplified ontologies. These are in form of topic taxonomies, typically used in knowledge management tools to organize the corpus and improve the user interface.

Furthermore, high-precision search is not intended to compete with general purpose search engines such as Google or Bing. High-precision search has a different role than a general purpose search engine. High-precision search is likely to be a niche product play in a specific knowledge domain that has a recognized need for: a) a higher level of precision not available from Google or Bing, and, b) the ability to define complex query criteria in natural language. Medicine is a good example of this.

For cognitive search products targeting a niche domain and high-precision search, a large and complicated ontology is likely required. Large, complicated ontologies, such as medicine, are expensive to develop since no automated approach is currently available. Lacking such an automated method means that the ontologies must be hand-built; this is typically laborious and expensive. The fact that many, large ontologies exist in the medical domain, but not in other domains, suggests the existence of a need unique to medicine that justifies the expense of developing these ontologies.

Therefore, successful adoption of cognitive-based NLP and search capabilities outside of medicine, i.e., areas which requires the development of new ontologies, must meet two criteria as follows:

1. The identification of one or more significant market niches for products that use cognitive NLP and search outside of medicine, along with the value-add benefits these products provide.
2. Achieve a significant reduction in the labor and cost of ontology development such that new products identified in #1 above are economically feasible.

The question is what level of cost reduction for ontology development is necessary for the cost-benefit threshold to be reached such that cognition-based approaches become economically feasible. This is an unknown because the cognition-based approaches are so new, and enable so many different, highly innovative products that traditional market survey methods to ascertain cost versus benefit become almost useless. It becomes a sort of "blue ocean" strategy question (see https://www.blueoceanstrategy.com) since it is highly unlikely that similar products will exist in the market. Consumers may have little or no idea on how to use these products, especially the most innovative ones, and hence will likely have difficulty ascertaining value. Thus a dearth of data makes it difficult to ascertain the cost-benefit threshold.

From an intuitive viewpoint, it appears reasonable to assume that a very aggressive amount of cost reduction is required for developing ontology data. This viewpoint is driven by the current state of the technology market. The current market can, in part, be characterized by the availability of free information, e.g., search, and free applications via open-source products. Hence, consumers tend to have an implicit need or expectation for very low or no cost data or products. This requirement extends to metadata, such as ontologies, due to the requirement for a very low cost-benefit threshold.

An alternative scenario can be posited that cognition-based NLP and search become monetized and offered as a low cost service for free. This could eventually create a commodity market for ontology data. But, since under this scenario ontologies become commodities, the market can be characterized by cost-based competition. Hence, even if the ontology has a monetized value, it exists as a commodity with price differentiation, and hence, experiences downward price pressures. Therefore, the labor required to develop ontologies must be quite low to successfully compete on the commodity market.

Hence, for the purposes of identifying a rough-order-of-magnitude goal for cost reduction, it is assumed that the cost reduction for ontology development needs to be in the 80%-90% range, or better. It appears reasonable that this level of cost reduction is needed before cognitive approaches are economically practical for widespread use.

The automated ontology learning was included in the SPS system architecture since it appears reasonable to assume that a major cost reduction is required for the development of new ontologies before cognition-based approaches can be economically feasible. The functionality, components, and interfaces for automated ontology learning are part of the system architecture. This ensures that these needs are taken into consideration as part of the research discussed in this dissertation.

1.6. RESEARCH STEPS

The research discussed in this dissertation began with the development of a SysML model for a fictitious semantic processing system, and was based, in part, on the knowledge gained from developing the medText prototype. The purpose was to prioritize PhD research activities via the identification of key search functions and components. A summary of the model is provided in Section 2.1.

The cognitive relevance measure was developed next. It is used as the distance measure for rank ordering documents returned by the search and identifying the most relevant. The cognitive relevance measure uses a topology-based algorithm that extracts neighborhoods from the ontology. These neighborhoods provide cognitive information from both a conceptual and contextual standpoint, and for this reason, are referred to as cognitive neighborhoods.

The source of ontology data is the Unified Medical Language System [8] available from the National Library of Medicine (NLM), which contains a large number of medical ontologies. Without these data the feasibility of this research would be significantly curtailed. The ontology used for this research is the SNOMED ontology, one of the medical ontologies in the Unified Medical Language System. It was selected due to the simplicity of the SNOMED relationships such that the ontology can be represented as a directed, acyclic graph, or DAG. A DAG structure is necessary to compute cognitive relevance measure in a way that mimics the feedforward nature of the knowledge base stored in the cerebral cortex.

These concept/context neighborhoods are extracted from both the search criteria and the corpora to be searched. A simple relevance measure was developed to identify the size of the overlapping cognitive covering space between each document and the search criteria. This relevance measure is discussed in Section 2.5. The draft paper in Appendix B on this subject provides the current version of the cognitive relevance measure along with further discussion of how these topological covering spaces represent cognitive neighborhoods.

Use of a concept/context-based approach for search, along with a completely natural language user interface, requires the ability to identify concepts in text. The MetaMap concept recognition tool [9-13], available from the NLM, was initially used to perform this task. The precision of concept recognition has a direct effect on search precision, and preliminary testing of the medText prototype indicated that the precision of MetaMap is insufficient for the extremely high level of search precision desired. Hence, research was pursued for improving the precision of automated concept recognition, and is discussed in Section 2.6.

Further details on the research in automated concept recognition and cognitive relevance measurement is described in the papers provided in Appendices B and C.

2. SUMMARY OF FINDINGS

2.1. SEMANTIC PROCESSING SYSTEM FRAMEWORK

The framework for a semantic processing system was developed to aid in the identification of key functions and components necessary for high-precision search. Documentation of this framework is in the form of a SysML model.

A detailed description of the framework is provided in the Semantic Processing System (SPS) framework document, attached in Appendix F. This includes identification of market needs, use cases, functions, components, and key interfaces, along with selected excerpts from the SysML model.

Four key challenges were identified from this architecture, shown in Table 2.1. These challenges appear to exist regardless of domain.

Table 2.1: Key Challenges Identified from SPS Architecture Model

Challenge	Description
1. Computation of semantic relevancy	Quantify intersection of cognitive neighborhoods between search criteria and document retrieved from corpora
2. High-speed processing	For computing topological covering space by a computational method faster than graph walking
3. Accuracy of automated concept recognition	Limitations found in MetaMap that limits precision of cognition-based search
4. Availability of low-cost ontologies	From requirement to pursue markets outside of healthcare, this is needed to monetize cognition-based search

This framework is applicable to any domain. While this dissertation may provide examples from medicine, the framework is not unique to medicine and results from research in semantic tools for medicine are equally applicable to any domain. Table 2.2 provides a number of examples for different knowledge domains. The value of a cognitive approach to information retrieval appears ubiquitous across knowledge domains.

Developing the SPS framework occurred at the beginning of this research, and was used to guide the research in terms of focus and requirements. By using a systems approach, as research progressed, it became clear that the goal of a cognitive approach to search required selection of a small set of cognitive theories applicable across the system, i.e., in the design of each component and interface, so as to effect successful integration. This need exists regardless of the knowledge domain being searched.

For example, the cognitive theory used for automated ontology learning likely influences the approach used to extract stated facts from text, where, in keeping with the use of a cognitive approach, a stated fact is a triple of concepts as follows:

$$fact = \{from, rel, to\} \tag{1}$$

$where$:

$from = subject\ concept$

$rel = relation\ type\ concept(equiv.to\ verb, modifier, adjective, etc.)$

$to = object\ concept$

Moreover, since a cognitive approach is used for search, use of a cognitive grammar approach is preferred for analyzing sentence and phrase meaning to extract the stated facts. The cognitive grammar approach is consistent with the triple shown in Equation 1. Note, however, that cognitive grammar is not a theory of cognition, but a relatively new approach in linguistics for grammatical analysis. Hence, automated approaches to linguistic analysis, such as those currently in use for computational linguistics, are not yet available.

In addition, text preprocessing is necessary to index the text with matching concepts found in the ontology. Like automated ontology learning, it also requires a component for analyzing sentence and phrase meaning using cognitive grammar. Since these components interface with utility component that performs text grammar analysis, a common theory of cognition is desired to ensure that the automated ontology learning and automated concept recognition components work in a manner consistent with the following:

- Grammar analysis
- Stated facts extraction and analysis
- Consistency in defining and using ontology concepts and relations.

2.2. NEED FOR AUTOMATED ONTOLOGY LEARNING

While the use of conceptual and contextual data has been shown to improve search precision [6], the availability of ontological data is a major stumbling block to the wide-spread use of cognition-based search. A large number of medical ontologies are freely available from the NLM [14], however, a dearth of ontologies exists for other domains. Moreover, developing ontologies is a manual process, and hence is expensive due to the large amount of labor required, typically involving high-dollar subject-matter experts. In addition to cost, the manual process used to create ontologies can create accuracy problems in representing a knowledge domain. This is mainly due to human error, for example, a subject matter expert neglect to include a concept and relationship.

A number of markets were identified other than healthcare that have significant revenue potential if automated ontology learning were available. The core performance requirement for automated ontology learning is 80% or better reduction in the cost of developing the ontology. Examples of these markets and products are shown in Table 2.2. In aggregate these markets and products may have a revenue potential up to $200MM per year.

Table 2.2: Example Products Enabled by Automated Ontology Learning

Example Market	**Example Product**
Knowledge Management	Application plugin or web service used by knowledge management application for automated taxonomy or ontology creation from corpora stored inside corporate firewall.
	Cognitive search tool for high-precision retrieval of documents inside the corporate firewall.

Table 2.2: Example Products Enabled by Automated Ontology Learning (cont'd)

Example Market	**Example Product**
Domain Specific, High-Precision Search	Web service, provided by a literature vendor (e.g., Amazon), to extract ontology from documents/books for a specific knowledge domain.
	Cognitive search of the knowledge domain, and in addition, an ontology-driven learning roadmap for self-learners. This product is in addition to the purchase price of the product, and hence offers a new revenue stream.
Product Literature Search	Similar to domain-specific search above, but provided as part of the help literature for a specific product, e.g., search tool for retrieving answers on a software product.
Systems Engineering	Domain documentation for user needs analysis.
	Requirements analysis for system and components.

Examples of the use of ontologies in systems and other engineering fields can be found in [15-37]. The market niche for these products can be summarized as follows:

- High-precision search tailored to one or more specific knowledge domains
- Automated ontology learning that provides a cost-effective approach to developing and maintaining the ontology for knowledge domains represented in a given corpora of text.

The current competitors of these products are any public-facing, internet-based search engine. None of the products in Table 2.2 can compete head-to-head with current internet-based search engines such as Google or Bing. In other words, to find relative straight-forward information, such as restaurant in the area serving a specific dish, use Google or Bing. If looking for specific information on a complicated topic, using complicated query criteria in the form of natural language, then a cognition-based approach would be preferred.

2.3. CURRENT APPROACHES TO AUTOMATED ONTOLOGY LEARNING

The major stumbling block to the products listed in Table 2.2 is the ability to develop low-cost ontologies. Researchers have been pursuing automated ontology learning for a number of years, but with limited results. None of the efforts: 1) were found to be effective enough to warrant wide-spread use, and, 2) are based upon a theory of cognition that enables both concept and context-based search. Ideally a method would be found in the literature using a general theory of cognition that is applied to the problem of mimicking human language cognition, for example confabulation theory. However, none of the approaches found used any theory of cognition other than adaptive resonance theory. A sampling of these efforts is shown in Table 2.3.

Table 2.3: Examples of Current Approaches to Automated Ontology Learning

Approach and References	**Limitations**
1. Recursive and non-recursive ART-based neural networks, in one case using a Bayesian network. Used either entropy or legacy term frequency-inverse document frequency (tf-idf) measures to optimize precision [38, 39]	• Uses measures inconsistent with cogency measure from confabulation, the chosen cognition theory used for this research • Lacks ability to provide natural language interface • Bayesian approach is not based upon a theory of cognition, and hence will not achieve improved search precision provided by a concept-based approach.
2. Extraction of smaller, focused ontology for a knowledge domain from larger ontologies [40]	• Assumes existence of larger ontology, which is contrary to the objectives of this dissertation, i.e., extract any ontology from exemplar corpora. • Lacks ability to provide natural language interface
3. OntoMiner, an application for creating ontologies from web pages [41]	• Level of knowledge desired in ontology typically greater than that found in web pages • Lacks ability to provide natural language interface
4. Extraction and tuning user input, where user input is set of common sense rules on a knowledge domain [42]	• Requires existing database of common sense rules from those familiar with knowledge domain, rather than learning from exemplar corpora for domain • Lacks ability to provide natural language interface

Table 2.3: Examples of Current Approaches to Automated Ontology Learning (cont'd)

Approach and References	Limitations
5. Use of Deep Neural Network, with architecture modified to use a convolutional layer. Uses a modified version of existing method for word encoding to account for word sense ambiguity [43]	• Appears to be much closer to the cognitive process described by confabulation theory, however, it was used for relation extraction only • Lacks ability to provide natural language interface
6. Navigli, *et al*. [44-46], lattice-based entity identification approach using an iterative extraction method and a number of the Onto series of applications (these do not appear to be in common use)	• Focuses on term extraction using a lattice structure, without extraction of relation types • Lacks ability to provide natural language interface

Among the approaches listed in Table 2.3, the use of neural networks, item #6, appears to hold the greatest promise for mimicking the cognitive process described by confabulation theory. The approach described by Chen, *et al*. [43] in item #6 includes layering and optimization typical of a deep neural architecture, which is similar to the multi-layer cognition described by confabulation theory.

2.4. ONTOLOGY AS NEURAL NETWORK

The current approach to store the ontology is typically a relational database. A portion of the ontology is loaded into working memory when required for computational needs. In theory the ontology can be represented as a neural network. Then, during text processing, the search tool accesses the required ontological directly from the neural network. This, of course, includes invoking the required cognitive functions.

This requires a neural architecture that has the ability to execute the confabulation cognition process as well as store the ontology in a structure suitable for use when executing cognitive processes. As shown in Figure 2.1, this includes the ability to store concepts (as neural codes) and ontology relationships (as knowledge links), including the ability to store relationships by relationship type (where a relationship type is a concept).

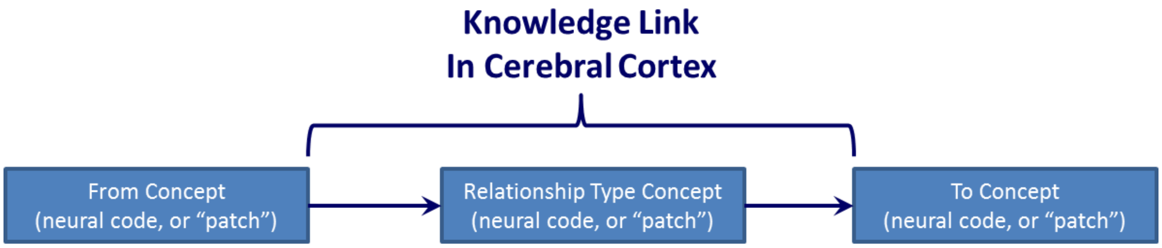

Knowledge Link
In Cerebral Cortex
From Concept
(neural code, or "patch")
Relationship Type Concept
(neural code, or "patch")
To Concept
(neural code, or "patch")

Figure 2.1: Stated Fact Triple as Knowledge Link: fact triples extracted from text define knowledge links that exist in the cerebral cortex. There is a 1:1 relationship between a knowledge link in the cerebral cortex, the stated fact triple extracted from text as shown in this figure, and the concept-relationship-concept triples in the ontology.

A summary of requirements for this neural architecture is as follows:

- The neural architecture provides long-term storage of the ontology, and it does this in a manner that mimics the cerebral cortex, i.e., concepts are stored as a group of neurons, and relations between neurons mimic the knowledge links of the cerebral cortex necessary for cognition. Furthermore:
  - These knowledge links represent the stated fact triples (Equation 1) extracted from text during the learning process.
  - A knowledge link must have a type, and the type must be the same as that identified in the stated fact triple, i.e., a knowledge link type is a concept.
- All concepts are single-word concepts. Relationships are used to identify compound concepts to reduce concept recognition computational requirements, as follows:
  - For example, the compound concept "spinal fusion" will exist in the ontology as two explicitly separate concepts. Knowledge links in the neural structure reflect that they are related, e.g., the knowledge link reflects the fact that a fusion can occur in the spinal region. In the SNOMED ontology, for example, some concepts are combinations of separate concepts and as a result have multi-word names to reflect this. In some cases these are multi-phrase names. This, in turn, requires

more sophisticated concept recognition approaches, which includes dealing with NP-hard combinatorics of word combinations (see Appendices A and B). A single-word concept reduces the complexity of automated concept recognition.

- The neural architecture and related system components provide the ability to calculate cognitive relevance, which includes:
    - The identification of cognitive neighborhoods for a set of concepts,
    - The identification of the intersection of two or more neighborhoods, and,
    - The calculation of the size of the neighborhoods and neighborhood intersections.
    - As a goal, ideally these are emergent properties of the neural architecture. For example:
        - The identification of cognitive neighborhoods is determined by which concepts and knowledge links are energized
        - The cogency of a possible outcome is represented by the level that a concept is energized, and the optimum is determined via a winner-take-all approach (such as on-center-off-surround).
- To enable text processing, long term storage must include the possible words and symbols encountered in text, and stored in a manner that enables cognitive processes.
    - That is, words should be stored as one or more neurons, along with one or more neurons for each character in the character set, and include relationships between the character set and each word. Similarly, the word associated with a concept should be stored as a group of neurons, along with a knowledge link to the characters that represent it.

- The neural architecture must provide short-term memory of text along with short-term storage of recognized concepts used for cognitive processing, to enable the cognitive processes of: a) ontology learning, b) accessing the ontology for concept recognition in text, and, c) determining cognitive relevancy.
    - This must include a process for recognizing sequential tokens in text and a cognitive process that mimics the human reading process, which includes recognizing characters, words, and finally concepts in a sentence and resolving word sense ambiguity.
    - This also includes a recurrent process of concept recognition as each new token in the sentence is added to short-term memory. This recurrent process includes recognizing the cognitive relation between new and prior concepts recognized. Hence the process includes use of prior sentences to improve the precision of concept and relation recognition, as well as sense disambiguation when required.

The scope of work necessary to perform research and to develop such a neural architecture, along with the associated software code and testing, is significant. This may be offset if the neural architecture simplifies or removes altogether the components and data currently in use for basic natural language processing tasks. A more important, and more general, benefit is that research in this area could make a significant contribution if it successfully mimics a significant portion of human language cognition.

2.5. COGNITIVE RELEVANCE AND SEARCH PRECISION

A topological covering space is defined for the ontology to identify cognitive neighborhoods. The size of the cognitive neighborhood is defined as the number of concepts in the covering space. The relevance of a document to the search criteria is defined as the intersection of the cognitive neighborhood for the search criteria and the cognitive neighborhood for the document. This is referred to as cognitive relevance, and is quantified as the number of concepts in the intersection of these two neighborhoods.

Topological covering spaces for the ontology are defined using the subsumptive property of the ontology, where the ontology is represented as an acyclic directed graph.

Subsumption is the “is a” relationship type in the ontology, that is, the parent-child relationship, and multiple “is a” inheritance is allowed, i.e., a concept can have multiple parents. Types other than “is a” is allowed, with no known limitation as to type. For example, if a medical procedure is performed on a certain organ or anatomy, then an anatomical concept is included in the cognitive neighborhood via a “located at” relationship. This in turn includes all of the “is a” ancestors related to that anatomical concept, by inheritance. Hence a concept can have a covering space related to its core cognitive meaning, and also have a covering space for related concepts not in its core cognitive ancestry.

No publications have been found that describe the same or similar approach. The closest approach is described in Schenker, *et al*. [47]. While the approaches in Schenker, *et al.*, use graph-based measures, such as “maximum common subgraph” to quantify graph edit distances as a proxy for similarity, there is no direct relationship to cognitive approaches. Given the initial success of the cognitive relevance measure, albeit based upon limited physician testing, and its consistency with cognition theory, the cognitive approach appears to be a superior method. Moreover, the need for using a common theory of cognition across components to optimize component reuse and integration supports the use of the cognitive approach to relevancy. The limiting factor, as discussed previously, is the availability of domain ontologies.

Further discussion is provided in a draft paper that will be finalized upon completion of testing by a physician, James Levett, M.D. Dr. Levett is a cardiothoracic surgeon and Chief Medical Officer for the Physicians’ Clinic of Iowa, Cedar Rapids, IA.

2.6. AUTOMATED CONCEPT RECOGNITION

The automated recognition of concepts in text is a requirement when using a cognition-based approach to search. As noted in Section 1.6, an existing tool, MetaMap, did not provide the desired level of accuracy. MetaMap was used as the baseline when researching alternative methods.

MetaMap has two distinguishing methods/properties: 1) concept recognition occurs for phrases, and 2), it uses a linguistic heuristic for scoring candidate concepts and identifying the optimum match. Ideally the concept recognition approach would include inter and intra phrase relationships, however this level of sophistication is the subject of future research (i.e., author's post-doctoral position at the NLM).

A cognition-based approach that improved the precision of automated concept recognition was identified and validated using gold-standard data from the NLM. This demonstrated the efficacy of cognition-based approaches as an alternative to the MetaMap linguistic heuristics for scoring candidate concepts. The MetaMap linguistics heuristic limits concept recognition to the phrase level only.

The first attempt to find an alternative to the MetaMap's linguistic heuristics was a straight-forward approach of a simple polynomial. This approach and results are provided in Appendix A. Although a polynomial function was identified that produced acceptable results, the combinatorics challenge was enough of a roadblock to make this approach impractical. Hence alternatives were sought.

The cognitive theory of confabulation was investigated along with a multi-layer perceptron to approximate a function used to compute a distance measure for ranking candidate concepts. The perceptron would be used to score each candidate concept found during the concept recognition process. This score is used to rank candidate concepts and select the best available. Training of the multi-layer perceptron was performed using gold standard data available from the NLM.

This approach was successful, obtaining 81% precision, which is a 5% improvement over best available benchmarks (MetaMap). Details of the approach and results are provided in the draft paper Inverse Cogency for Concept Recognition (Appendix B).

The approach used for automated concept identification is conceptually straight-forward, as follows:

1. Identify noun phrases in a sentence using part-of-speech tagging.
2. Develop list of candidate concepts for each phrase.
3. Score each candidate, then rank candidates based upon this score to identify the best match.

Concept recognition was done at the phrase level for consistency with the gold-standard test data provided by the NLM. However, analysis of these results indicated a loss of fidelity of concept recognition. Based upon these results it was determined that a cognitive-based approach is needed to identifying inter and intra-phrase relations that indicate how concepts can be combined in a way to identify candidate concepts of higher fidelity. Combining phrases results in one or more concept triples, as defined in Equation 1. These triples are the cognitive artifact used to identify candidate concepts of higher fidelity. They are also the cognitive artifacts used to build the ontology.

Combining phrases and combining words within a phrase requires an analysis that identifies cognitive relations between words within and across phrases. In traditional linguistics the role of a word can be identified via part-of-speech tags. But these roles have no direct relationship with a cognitive approach to relation recognition necessary for an automated approach that improves concept recognition fidelity.

As an example of what a cognitive relation may look like, consider certain parts-of-speech, such as verbs, modifiers, and adjectives, which can imply a cognitive relation depending upon the context in or across phrases. Take the simple phrase "lumbar surgery." Surgery is the noun, lumbar is the adjective. From a cognitive viewpoint, lumbar is an anatomical location, which implies a relation, that is, a surgery located in the lumbar region, where in this case the relation is of type "located at."

However, an automated tool was not found that that provides this capability using a cognitive approach. The theory of Cognitive Linguistics [48], and more specifically its subtopic, Cognitive Grammar [49], was identified as the theoretical basis for identifying cognitive relations between words and phrases. However, an automated tool to perform this task, analogous to an automated part-of-speech tagger, is not available. The theory of Cognitive Linguistics is relatively new and still lacks the explicit, formal specifications necessary for reducing it to an automated approach.

Despite the relative lack of maturity of Cognitive Linguistics in comparison to legacy linguistics approaches, Cognitive Linguistics may offer promise due to its basis in cognition. For example, although Cognitive Linguistics/Grammar is relatively new, it has been used successfully in research for the automated identification of semantic annotations for Wiki pages [50].

Extracting cognitive relations in a sentence is an important part of the Semantic Processing System. Each cognitive relation identified in a sentence also identifies an ontological relationship between concepts. Hence, identifying inter- and intra-phrase relations using a cognitive approach addresses not only the issue of improved concept recognition fidelity, but also addresses the need to extract stated facts as part of the automated ontology learning process. Both cases result in identifying one or more stated facts in a sentence.

However, a significant amount of work is required to design and develop an approach to relation extraction using Cognitive Grammar, for the following reasons:

1. A standardized set of cognitive "tags" for the cognitive role of words does not yet exist in Cognitive Grammar. While a set of grammatical roles, e.g., part-of-speech, exist in existing linguistics theory, no such equivalent set exists for Cognitive Grammar.
2. No automated tools exist for the analysis of phrases and sentences and the assignment of standardized cognitive roles to words.

Correspondence with members of the International Cognitive Linguistics Association was conducted to validate the applicability of an automated approach to Cognitive Grammar, and to identify the approximate scale of research effort to develop them. Responses to queries validated that the cognitive approach to linguistics, in relation to ontologies, is applicable and valuable. An email from Dr. John Barnden, University of Birmingham, Department of Computer Science stated "…a general concern with concepts and their relationships in cognitive linguistics as a whole fits with ontologies…" and that the research is "plausible." Dr. Andrew Gargett, also of the Department of Computer Science at Birmingham, suggested approaching the International Computer Science Institute at the University of California, Berkeley, also pursuing research in this area. Dr. Gargett also provided input that such a research project has value, was "interesting," and "sounds plausible." However, it is not something he would recommend to a PhD candidate, unless the person has a "deep background in formal linguistic methods. And even with that background the project would be challenging."

In summary, researchers in the field concurred that use of Cognitive Linguistics has significant potential in the automated identification of inter- and intra-phrase cognitive relations, but the scope is beyond that appropriate for a PhD candidate. For these reasons research in the application of Cognitive Linguistics to the problem of identifying cognitive relations was not pursued further, and is deferred to future research. Deferment to future research does not imply a lack of importance; rather, it reflects the lack of practicality in pursuing it as part of a PhD candidate's research.

2.7. SYSTEM SUMMARY – COGNITIVE SEARCH MATURITY AND PROBABLE TRENDS

Results from research on concept and context-based search [6], even at a fairly rudimentary level, provides evidence that opportunities exist for significantly higher levels of precision.

However, cognition-based methods are relatively immature in comparison to legacy approaches currently in use. A systems view of cognition-based methods is necessary to mature the approach in a way that does not inflate development labor.

Results of the research described in this dissertation indicated that striving to improve search precision using cognitive methods inexorably leads to the use of methods that mimic human cognition. In the long run, mimicking human language cognition appears to provide the least-cost path to improved search precision across knowledge domains. Even though cognition-based methods are relatively immature at this time, their pursuit appears to provide significant opportunities in reducing net development labor by reducing the effort necessary to develop an integrated suite of natural language processing components.

For example, the labor required to develop the straight-forward infrastructure components for basic text processing in a Semantic Processing System can be significant. None of the steps for basic text processing are vastly complicated, but there are a surprising number of these small actions that add up, and they are highly related. In some cases surprisingly simple requirements, such as consistency in parsing text across components, can become problematic when components are open-source from different organizations that happen to use slightly different algorithms.

Consider the example of an open-source software library from Stanford for parsing text and performing part-of-speech tagging that was employed due to its ease of use. After integrating the Stanford tool into the research proof-of-concept application, gold-standard text annotations for testing the precision of automated concept recognition was made available by the NLM. These gold-standard data were developed using the NLM's in-house tools, which had simple differences in the start and end position of a text annotations, in some cases by just one character. It also had differences in assigning part-of-speech tags, which created random errors in results during testing using the NLM gold-standard data.

Hence, due to these problems it became necessary to re-write a number of components in the research proof-of-concept application (created to test the algorithms developed as part of this dissertation research). This created significant delays in research progress. The scope of work necessary for this conversion included removing the Stanford tool, becoming familiar with the NLM open-source components, and then integrating the NLM open-source components into the application. Due to the lack of documentation the integration of the NLM tool required significant trial-and-error (the reason for choosing the Stanford tool in the first place). Hence, modifications to use the NLM components amounted to a significant amount of labor for debugging and coding.

Therefore, the choice of open-source software to perform certain functions, as a tactic to reduce development cycle-time, can drive expensive changes to the system architecture simply due to integration. Indeed, use of open-source software to avoid development of a sophisticated component becomes both a blessing and a curse when it comes to trading the benefits of reduced development cost and timeline against the risk of incompatible integration.

The sensitivity of development labor versus improved precision also increases as the number of components increases, and the sophistication of these components increases. Examples include components and data for lexicons, ontologies, linguistic and grammar tools that identify morphological variants of words and determination of a common base, acronyms, abbreviations, part-of-speech taggers, and sentence boundary determiners.

Hence the return on investment question is posed when making system architecture decisions under constrained resources. A case can be made that it may be a better use of resources to focus on developing computational intelligence approaches resulting in the highest level of cognition and do this in a way that reduces, instead of increases, the need for the plethora of text processing components.

The research described in this dissertation tends to support this assertion – focus on computational intelligence methods that, for example, minimize dependence upon (or avoids altogether) the legacy linguistic analysis tools and data, such as lexicons and word variant databases and algorithms for word morphologies.

The human brain has none of these artifacts per se. Lexicons, linguistics, grammar, and other elements used in natural language processing are not separate structures in the brain. Artifacts such as lexicons and word morphology data used in natural language processing are physical representations of the properties of the human neural structure and knowledge links.

The concept of moving towards a purely cognitive approach is not new. Language experiments conducted for confabulation theory support this. Sentence completion experiments produced plausible and logical sentences without the need for a lexicon, ontology, or linguistic tools [1-3] . Hecht-Nielsen takes this viewpoint in his research on cognition, stating that “linguistics is an emergent property of confabulation” [3].

These language experiments, however, allow a random choice of topics and stated facts due to the random selection of words to complete a sentence. Practical application of the theory is typically constrained by other, additional requirements, requirements that did not constrain these language experiments. Hence, the avoidance of legacy tools and reference data, such as lexicons or part-of-speech tags, is a long-term goal to be achieved via mimicking human language cognition, and hence, likely not be achieved in the near future.

3. CONCLUSIONS

3.1. AUTOMATED CONCEPT RECOGNITION

The application of confabulation theory successfully improved the precision of automated concept recognition. This demonstrated that the inverse cogency measure, a modified version of the confabulation cogency measure, provides a reasonable distance measure when rank ordering candidate concepts.

Hence the rank ordering provided by inverse cogency, when used in combination with a multi-layer perceptron as function approximator, resulted in improved precision for concept recognition. Its use resulted in a 5% improvement over the best known baseline (MetaMap).

The algorithm for selecting candidate concepts and computing the inverse cogency has room for further improvement. For example, it does not yet include the use of synonyms and acronyms. This is fairly easily resolved and when fixed the precision of concept recognition is expected to improve further. Over the long-term, the identification of relations between concepts will also improve the precision of automated concept recognition, discussed in Section 3.4.

3.2. COGNITIVE RELEVANCE OF RETRIEVED DOCUMENTS

Two cognitive-based distance measures were developed to both enable concept-based search and to improve the precision of search. Neither measure achieved the desired results.

The approach used a simple-first philosophy, that is, try the simplest approach first and if that doesn’t work then increase sophistication incrementally. The poor performance of these two measures suggests that use of straight-forward ranking measures to improve search precision is inadequate.

With simple measures being inadequate for high-precision concept-based search, a more sophisticated neural network approach appears to offer greater promise. A neural network approach, if implemented correctly, may be more advantageous for reasons other than precision alone, principally in terms of maintenance labor.

Further details can be found in Appendix C.

3.3. SEMANTIC PROCESSING SYSTEM ARCHITECTURE

The system architecture for the Semantic Processing System provides a framework for identifying research direction and priorities that optimize the system as a whole. Lacking such a framework increases the difficulty in decision-making when performing trade-offs between potential short-term tasks and technology selection versus long-term goals. Lacking this framework also increases the risk of making decisions in a vacuum with the potential downside being suboptimal performance or lack of ability to integrate components.

Examples of the benefits experienced so far include:

- The selection of confabulation theory and its successful use to improve concept recognition precision
- The identification of cognitive linguistics as a potential approach to: 1) improve concept recognition, and, 2) extract stated facts from text as part of an automated ontology learning process.

The Semantic Processing System Architecture provided valuable insight into how decisions on component functions and design impact the long-term effectiveness of an operational system. It aids in the development of a long-term planning that, in effect, becomes a planning tool, roughly analogous to a sequenced research pipeline. Short and long-term efforts are linked to a common, long-term objective.

3.4. COGNITIVE LINGUISTICS

Use of Cognitive Linguistics, in combination with the use of inverse cogency and confabulation theory, holds the most promise for extracting stated facts from free-form text. Researchers in the field concurred with using Cognitive Linguistics, but cautioned that the scope of such research is outside that practical for a candidate PhD. Hence the application of Cognitive Linguistics is deferred, but as the recommended approach for future research to address the problem of extracting stated fact triples from text.

The extraction of stated fact triples is a key capability due to the requirement for this functionality to both improve the fidelity of concept recognition and to perform automated ontology learning.

3.5. AUTOMATED ONTOLOGY LEARNING

The financial success of cognitive-based search, with niche products in multiple markets, depends upon the ability of an automated ontology learning approach to reach an 80%+ reduction in the cost of ontology development. While the ideal level of manual effort is 0%, it is assumed that some minimum amount of manual labor is required for practical reasons. Hence, the objective is stated as an 80% or better reduction in labor.

The automated ontology learning process is expected to use the following steps:

1. Calculate cogency values required by confabulation theory, i.e., analyze a corpora of text representing the body of knowledge, perform word counts, and calculate the conditional probabilities of word combinations
2. Use an iterative process to build the ontology – during each iteration of the process, in combination use Cognitive Grammar and inverse cogency to iteratively build the ontology. The iteration occurs word by word in each phrase of a sentence, as follows:
    a. Extract stated facts from free-form text
    b. Analyze each stated fact to determine if it can be added to the ontology; it cannot be added if any of the following conditions exist:
        i. The triple is a duplicate, or a triple using the same two concepts already exists but using a different relation concept
        ii. Lexical analysis – word morphology, acronym, abbreviation, etc. using domain lexicon – indicate that the concept already exists using a different word form
        iii. The triple creates a cycle in the graph – it must be acyclic
    c. Add new stated fact triples to the ontology
        i. If at beginning or end of a cognitive neighborhood, add to extend neighborhood

        ii. If in middle of neighborhood, insert into existing path as appropriate

    d. Continue to the next phrase

It appears that the optimal architecture for the ontology and related cognitive processes is via a neural architecture discussed in Section 2.4. The cognitive theories to use in this architecture most likely includes confabulation theory, although with further research additional theories may be found that are useful.

The development of an automated ontology learning capability is deferred to future research for the following reasons:

- The amount of effort to develop an automated ontology learning process is significant.
- It is dependent upon the availability of recognizing stated fact triples in text, which is deferred to future research

3.6. SUMMARY – CONTRIBUTIONS AND REMAINING CHALLENGES

Results from research on concept and context-based search [6] provides evidence that opportunities exist for higher levels of precision using a cognition-based approach.

The research described in this dissertation has developed two concrete approaches that advance the state-of-the art in cognitive-based search, as follows:

1. The application of cogent confabulation to improve the precision of concept recognition in text. Concept recognition is a necessary task for cognitive-based search that impacts search precision. The approach to automated concept recognition discussed in this dissertation includes the development of:
    a. A modified version of cogency for use with ontologies, called inverse ontology cogency, and,
    b. A multi-layer perceptron approach that approximates a function used to rank candidate concepts and identify the candidate of highest precision.

    To the author’s knowledge a measure like the inverse ontology cogency does not yet exist in the literature.

2. Development of an innovative topology-based approach to improve the precision and ease-of-use of cognitive search for complex queries. The approaches discussed in this dissertation includes the development of:
    a. A cognitive neighborhood for a set of concepts, and,
    b. A cognitive relevance measure, which makes use of the intersection of these neighborhoods, used to rank candidate documents returned by the search and identify those having the highest cognitive relevance to the search criteria.

    To the author's knowledge no covering space approach or measure defining cognitive relevance in this manner has been previously developed.

The research described in this dissertation also identified challenges that must be overcome before cognitive-based search can be practical and adopted in wide-spread business use. This analysis is from the perspective of a new or existing business determining the likelihood that such an approach can be successfully monetized. These challenges include:

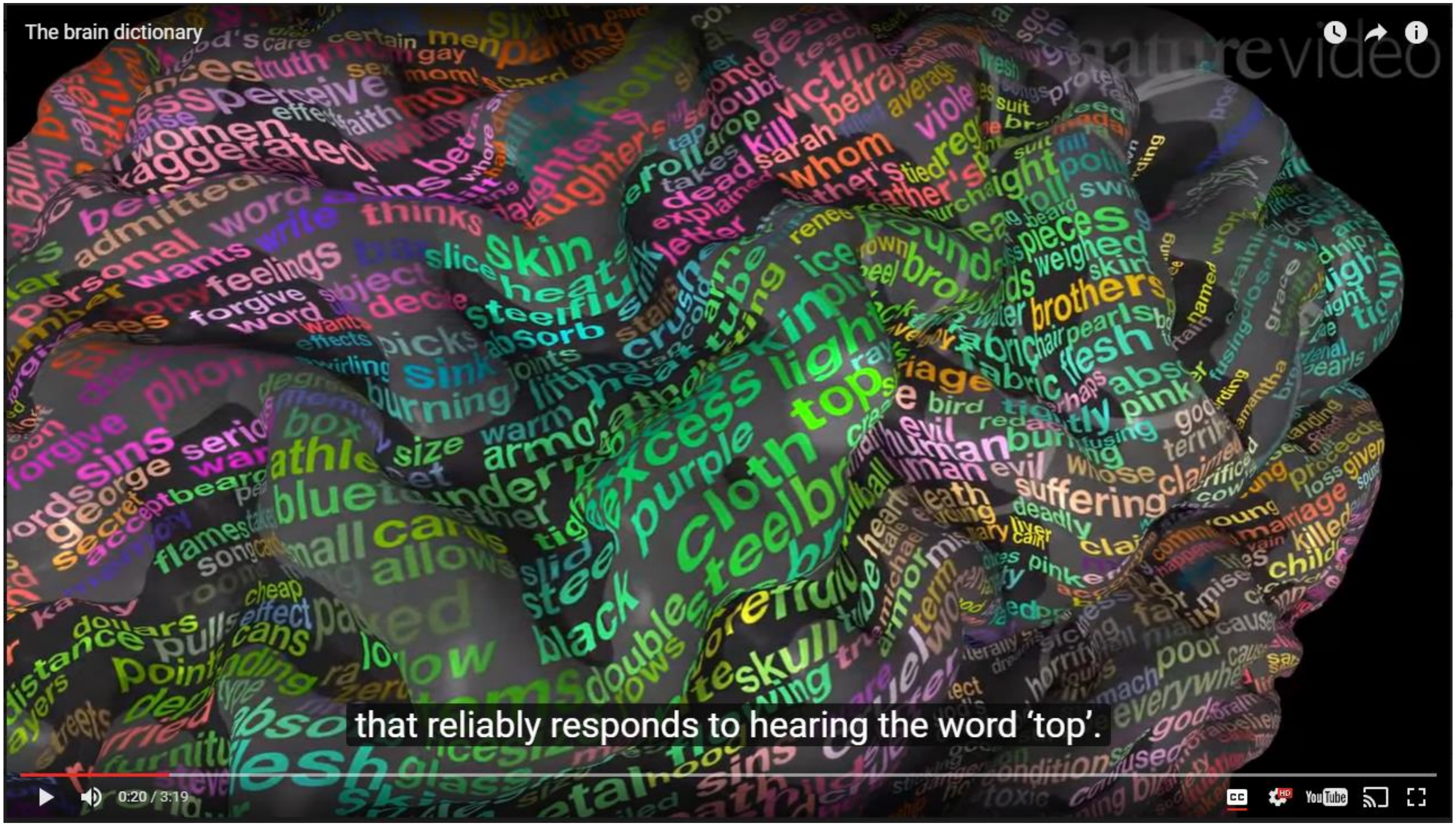
1. <u>Use Cognitive Linguistics/Grammar</u>: An approach that automates the analysis of text to identify relations between concepts found in text, i.e., the extraction of stated facts. The use of the theories of Cognitive Linguistics, possibly including Cognitive Grammar, was identified as the most promising for this task. Extracting stated facts from text is needed to:
    a. Improve the fidelity of automated concept recognition, and
    b. Perform automated ontology learning.
2. <u>Use Single Word Ontology</u>: This potentially minimizes the complexity of the concept recognition task.
3. <u>Increase Computational Intelligence Value-Add</u>: As the complexity of NLP grows in the pursuit of improved precision, the resources expended may be driven by basic NLP infrastructure efforts, rather than developing value-add cognitive methods for improved search.

    Ideally this growth in labor can be addressed with cognitive methods that avoid the expense of traditional linguistic tools. For example, analysis of corpora of text, such as news articles, was all that was necessary for language completion experiments conducted for confabulation theory. No lexicon or linguistic tools were required.

4. Ontology as Emergent Property of Cognition: Another long-term alternative to consider is the development of a neural architecture that stores an ontology and performs concept recognition and search tasks described in this dissertation. This is a significant amount of work. If successful, however, it may help avoid the use of the plethora of linguistic tools and databases typical of cognitive-based search. This approach appears to be consistent with the most recent findings in neurological research, notably confabulation, as discussed previously, and also the neural-word mapping found in the cerebral cortex [52].

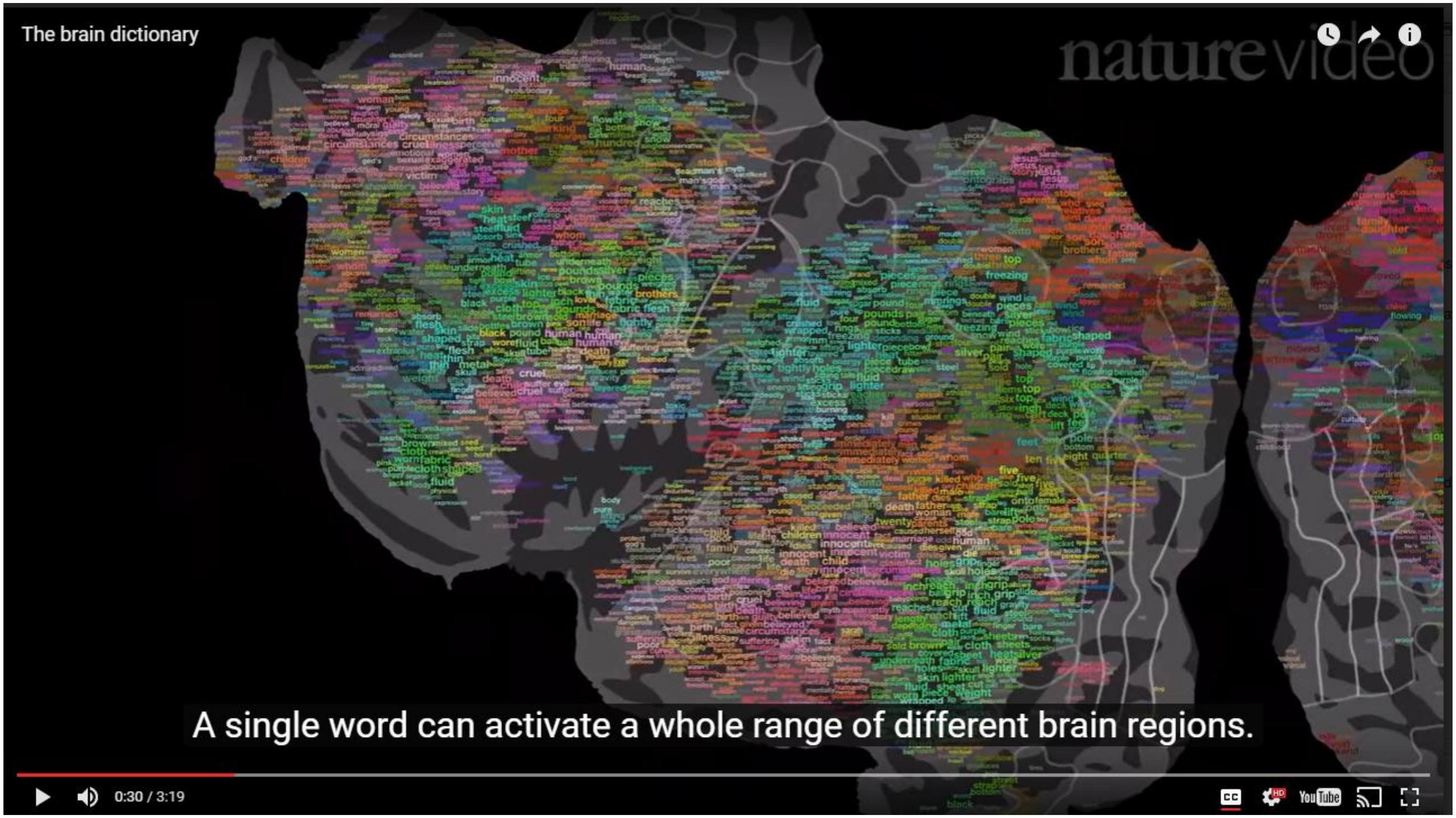
In addition to confabulation theory, other research supporting the notion that the ontology is an emergent property of cognition can be found in Huth, *et al.* [52]. This research identified the map between words and storage locations in the cerebral cortex for these words using fMRI linked temporally to a story read to the study participants. Word locations in the cerebral cortex were similar across study participants. In addition, evidence indicated that the conceptual meaning of a word was stored in the cerebral cortex. Confirmation of storing the conceptual meaning of a word was obtained using words that had multiple meanings depending upon the contextual use (i.e., different word senses). Different word senses for the same word had different storage locations. In addition, words similar in semantic context are co-located in the cerebral cortex, presumably to minimize the latency of accessing semantically similar concepts. A sample of word locations is provided in Figure 3.1. The map for the entire cerebral cortex is provided in Figure 3.2.

Storing concepts instead of words and the co-location of semantically similar concepts in the cerebral cortex appears to support the notion that the ontology is an emergent property of cognition. The ontology consists of a set of concepts (not words), and the ontology has topologically co-located neighborhoods for concepts that are cognitively related. The topological co-location of concepts is in regards to the cognitive neighborhoods found in the ontology. These neighborhoods are defined by the ontological relations between concepts. Hence the storage and access of an ontology using biologic mimicry, i.e., neural networks, appears to be a reasonable approach (see draft paper Cognitive Relevancy in Appendix C for further discussion).



that reliably responds to hearing the word 'top'.

Figure 3.1 Example of Cerebral Cortex Word-Concept Map: This is an example mapping between words and location that the corresponding concept is stored in the cerebral cortex. Screenshot from YouTube https://youtu.be/k61nJkx5aDQ.



A single word can activate a whole range of different brain regions.

Figure 3.2 All Word-Concept Maps in One Hemisphere of Cerebral Cortex: This displays all word-concept storage locations for one hemisphere of the cerebral cortex. Screenshot from YouTube https://youtu.be/k61nJkx5aDQ.

APPENDIX A

OBJECTIVE FUNCTIONS FOR TEXT CONCEPT TAGGING

This paper was published in 2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology.

Objective Functions for Text Concept Tagging

Proof of concept for GP approach that identifies objective functions for automated tagging and its potential role in bioinformatics and knowledge acquisition

George J. Shannon, Ph.D. Candidate
Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, Missouri, USA
gjscnc@mst.edu

Donald C. Wunsch II, Ph.D. EE, MBA, PE
Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, Missouri, USA

dwunsch@mst.edu

Steven Corns, Ph.D.
Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, Missouri, USA
corns@mst.edu

***Abstract*—Bioinformatics can involve semantic information extraction to retrieve knowledge found in free-form text. Examples of these sources can include research literature, textbooks, or clinical notes in electronic health records. Performing semantic search, that is, searching for concepts instead of keywords, necessitates that the free-form text be tagged with the ontology concepts that best matches each word or phrase in the text that is linguistically meaningful. For computational intelligence applications this employs a tagging process that automates this general approach, e.g., select candidate concepts, typically a large set of candidate concepts extracted from the medical ontology, and then rank the candidates using an objective function which quantifies tag accuracy. Tag accuracy is defined as the accurate match between the ontology concept and the linguistically meaningful words and phrases found in the text being tagged.**

This research is a proof-of-concept on the use of genetic programming to derive the objective function which ranks the candidate concepts and selects the set of best matching concept for a sentence. A short set of example primitive and linguistic variables are used as input to the GP process, and a set of manually tagged sentences extracted from the literature is used to derive different objective functions potentially suitable for tagging. This proof-of-concept demonstrates the potential of this approach to simplify automated semantic tagging, and also to identify some of the challenges likely encountered when applying the GP approach to a complex linguistics problem of this nature.

Keywords*—semantic text tagging; genetic programming, natural language processing, computational intelligence

I. INTRODUCTION

This paper will present results for a proof-of-concept for tagging text using a genetic programming (GP) approach. It differs from prior approaches by making no a priori assumptions on the objective function used to score candidate concepts to rank and select the best matching concepts used to tag text.

As a means to highlight the relevancy and importance of this research the paper will present results within the context of knowledge acquisition as a typical task involving bioinformatics and computational intelligence. We will present how the GP approach indicates that the potential exists for simplifying certain natural language processing (NLP) tasks typically found with text tagging. Differences between the new GP approach and an existing approach (MetaMap from the National Library of Medicine (NLM)) will be highlighted. Most notable is the tagging of text at the sentence instead of phrasal/part-of-speech level.

This paper begins with an overview of the fundamental purpose of concept text tagging and its role in knowledge acquisition as it relates to bioinformatics. While this can be a somewhat didactic step, it is included to communicate the motivation for the research, that is, that text tagging is a fundamental aspect of concept-based search since any shortcomings with text tagging flow through the semantic processes. After this overview then the GP approach and results are presented. This includes a review of the MetaMap existing approach whose basic linguistic heuristics were leveraged for use in the GP process. Since GP is a legacy computational intelligence approach, this paper will not focus on the details of the GP to a great extent since GP is so well known and no particularly innovative approaches were taken with GP per se. The main contribution is determining the feasibility of using GP to evolve an objective function that can accurately tags real-world medical text. Proving, extending, or innovating the approach beyond the feasibility stage is left for future research.

II. OVERVIEW AND BACKGROUND

A. Motivation

1) Concept-Based Search

Fundamental to all research is the objective of concept-based search, which we refer to as semantic search synonymously. One example of this is the “Bag of Concepts” search, in comparison to a traditional “Bag of Keywords” [1]. (The approach in [1 is provided as an example only. It used a Support Vector Machine approach for mapping words to concepts, along with part-of-speech tagging. We are striving to eliminate part-of-speed tagging to reduce complexity.)

Concepts in medical domains are multi-word phrases and can be complex. One concept can contain multiple words and compound concepts. As an example, take the concept “Dorsolumbar spinal fusion with Harrington rod” from the SNOMED-CT ontology (part of the Unified Medical Language System (UMLS) [2]). From a keyword indexing perspective this adds complexity due to the requirement to process multiple keywords for what is a single concept, and then compile results based upon a vector of individual keywords likely to be returned by a traditional keyword search engine. In comparison, a concept-based search is looking for one concept, not multiple keywords.

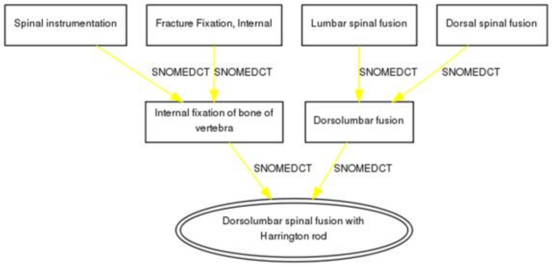


FIGURE 1: CONCEPT-BASED SEARCH EXAMPLE - a single concept can represent what is typically multiple keywords for a medical concept, potentially improving accuracy of information retrieval. This figure from the UMLS Terminology Services web site [3].

As shown in Figure 1 concept-based search can enable the use of ontology context information to enhance the accuracy of results. Prior work by one of the authors (unpublished) involved the creation of a prototype of a semantic search engine for medical text. Covering space heuristics, based upon simple topology theory, was developed for this proof-of-concept; this approach appeared to significantly improve the accuracy of the search results (details are outside the scope of this article).

However the use of concept-based search is not a panacea. Part of the motivation for research in tagging approaches is that concept-based search obviously does not remove search complexity altogether. The complexity of using multi-keyword searches is not removed but in effect replaced by the complexity of NLP, one aspect of NLP being the need to tag text with corresponding concepts. To a large extent the motivation for the research we are conducting is to minimize this complexity.

2) Extend/Simplify MetaMap

One of the approaches to this research is to investigate opportunities to enhance the capabilities of an existing tool currently in use for semantic text tagging of medical literature. The semantic search prototype used the MetaMap [4-6] tool from the National Library of Medicine for tagging literature and search criteria with concepts found in the UMLS. This was chosen as the starting point for tagging due to the success of MetaMap.

Experience gained from developing the aforementioned semantic search prototype suggested that opportunities may exist to enhance the accuracy, simplify processing, and reduce complexity of text tagging performed by MetaMap. The process used by MetaMap can be summarized in Figure 2 as follows:

1. Parse document into paragraphs and sentences.
2. Parse each sentence using part-of-speech tagging (POS) and retain noun phrases.
3. For each noun phrase in sentence extract list of candidate concepts from the UMLS ontology.
4. Score each candidate noun phrase using linguistic rules.

FIGURE 2: BASIC METAMAP PROCESSING STEPS – note step 3 where concept tags are computed at the noun phrase level not the sentence, making it necessary to define a heuristic to compile concepts across phrases for a single sentence, which may add complexity not required for a particular semantic application.

One area of interest is step #3 – parsing the sentence into part-of-speech elements and matching the concept tags to each noun phrase. To use MetaMap with the semantic search prototype it was necessary to assimilate the tags for a sentence from the tags for all the noun phrases, and in turn for a document assimilate tags for all sentences. Since visibility down to the noun phrase level wasn't needed an opportunity appeared to exist for developing an approach that tagged text at the sentence rather than at the noun phrase level.

3) Reduce Tagging Computational Complexity

Included in potential enhancements to the existing MetaMap process mentioned in the prior paragraph is the possibly of simplifying the tagging process, specifically by removing the requirement for part-of-speech tagging (POS). If tagging is performed at the sentence rather than noun phrase level then it appeared that the need for POS may be circumvented completely along with its incumbent bandwidth load.

Not only would removing the POS requirement potentially reduce bandwidth requirements and application complexity, it could enhance the flexibility of the approach. This could occur since removing the requirement for POS may also remove uncertainties in tagging outcomes associated with tailoring or training a POS tool for a particular knowledge domain. Hence a GP approach creates an objective function for tagging at the sentence level could circumvent any potential stumbling blocks associated with identifying a suitable existing POS tool, or building a new one, and training it for a use with a specific domain. Among the potential application areas are the Semantic Web [7-9], requirements analysis and architecture [10-16], ontology learning [8, 9, 17], and ontology/concept-based learning [5, 18-20] to cite a few examples.

B. Context – Systems View of Knowledge Acquisition

Viewing the need at a higher 'systems' level, bioinformatics can involve semantic information extraction to explore existing knowledge found in free-form text. This may involve hypothesizing new knowledge from that gleaned from existing. Below is a potential knowledge acquisition process that conducts cycles of information extraction iteratively, hypothesizing new other knowledge/conclusions as new knowledge is gained from each successive cycle.

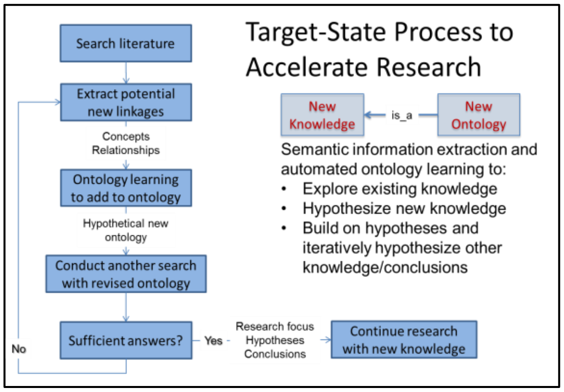

Target-State Process to Accelerate Research
Search literature
Extract potential new linkages
Concepts Relationships
Ontology learning to add to ontology
Hypothetical new ontology
Conduct another search with revised ontology
Sufficient answers?
No
Yes
Research focus Hypotheses Conclusions
Continue research with new knowledge
New Knowledge
is_a
New Ontology
Semantic information extraction and automated ontology learning to:
• Explore existing knowledge
• Hypothesize new knowledge
• Build on hypotheses and iteratively hypothesize other knowledge/conclusions

FIGURE 3: POTENTIAL KNOWLEDGE ACQUISITION PROCESS – this highlights the importance of text tagging within the overall approach to bioinformatics, where the artifacts of new knowledge are new or revised domain ontologies.

In the Figure 3 process, the ontology is the artifact that documents the acquisition of knowledge. Revisions, additions, or completely new ontologies provide the artifacts of new knowledge used in computational intelligence. Semantic information extraction is part of the process that extracts new knowledge facts from literature.

A semantic processing system, or SPS, for semantics-related functions for knowledge acquisition can be decomposed into a small number of basic components as shown in Figure 4. This system is hypothesized to be the system infrastructure common to most semantic processing systems.

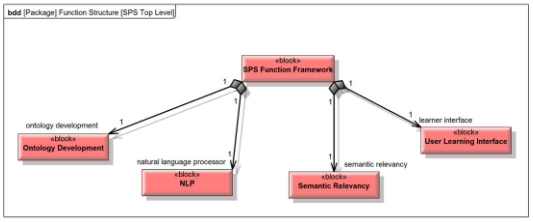


FIGURE 4: A SEMANTIC PROCESSING SYSTEM (SPS) FRAMEWORK - basic components of a semantic computational system common to most applications

When conducting a literature search the SPS component functionality invoked includes NLP and semantic relevancy (relevancy component provides quantification of results and ranking results). Ranking results, specifically, quantifying how closely the search results match search criteria, requires the text being searched be tagged with matching concepts, i.e., words and phrases are matched to the appropriate concept found in the ontology.

Further details on the role of ontologies and automated ontology learning to aid in semantic search is beyond the scope and focus on this paper (the role of ontologies to enhance semantic search was demonstrated by the prototype of the semantic search engine described earlier).

However it is clear that concept tagging is a fundamental and important step for accurate information retrieval when using semantics, i.e., concepts, as the search criteria. The basic context of knowledge acquisition was presented here to highlight the fundamental role of semantic information extraction and demonstrate that semantic tagging is an enabling technology. The speed and accuracy of semantic tagging has an impact on the speed and accuracy of knowledge acquisition and computational intelligence activities.

C. MetaMap 'Objective Function' and GP

When MetaMap performs tagging functions it applies a set of linguistic heuristics [21] that scores each candidate concept. The highest scoring concepts are considered the optimum tags for each noun phrase. A summary of the MetaMap linguistic equations is as follows (see [21] for details and examples):

$$Score = 2(cenrality + variation) + coverage + cohesiveness \tag{1}$$

$where$:

$$centrality = 1\ if\ \ string\ involves\ head\ of\ phrase,\ 0\ otherwise, \tag{2}$$

$$variation\ v = {}^{4}\!/_{(d+2)}, \tag{3}$$

$$where\ d = morphological\ distance$$

$$coverage\ c = {}^{2}\!/_{3}\,(concept\ span) + {}^{1}\!/_{3}\,(phrase\ span) \tag{4}$$

$$where\ span = {}^{\#words\ matched}\!/_{\#\ words\ in\ string} \tag{5}$$

$$cohesiveness\ coh\ = \textstyle\sum_{i=\#components} length^2_{component_i} \Big/ length^2_{string} \tag{6}$$

$$coh_{total} = {}^{2}\!/_{3}\, coh_{concept} + {}^{1}\!/_{3}\, coh_{phrase} \tag{7}$$

Coverage (4) is a measure of how many words in the concept are covered in comparison to words covered in the noun phrase and cohesiveness (6) is a measure of contiguous words. When calculating cohesiveness, $component_i$ refers to each contiguous segment of text, where the contiguous segment may be in the concept or in the noun phrase string being matched.

These linguistic approaches were evaluated and certain aspects included in the GP approach to determine if a new objective function could be derived that does not require POS tagging such that tagging can occur at the sentence level. A minimum set (i.e., Occam's razor) approach was taken to strive for the least complex function.

III. GP AND DERIVING OBJECTIVE FUNCTIONS

The purpose of the GP approach is to explore tagging at the sentence level that reduces computational complexity while maintaining or improving tag accuracy. This approach targets situations where visibility of concept tagging at the phrasal level is not required (i.e., when only needing the Bag of Concepts).

Each 'tag' for a sentence is a set of candidate concepts having at least one word in the concept match one or more words in the sentence.

$$tag_{candidate} = \{C_1, C_2, C_3, \ldots C_i\} \tag{8}$$

$$where\ C_i\ is\ the\ concept\ mapped\ to\ sentence\ word\ i$$

Note that concepts can repeat, that is, each $tag_{candidate}$ is not a mathematical set in the formal set (intersection, union, etc. laws do not apply since a concept can appear multiple times in a sentence). However, the relationship between an individual word in the sentence and the tagged concept is one-to-one, that is, each position in the sentence can be tagged with either 0 or 1 concept.

A match occurs when any linguistic base word is matched, i.e., the word in the sentence and word in the concept are morphological variants of the same base word. These variants are available from the UMLS lexicon database [22]. Since this initial research is for a proof-of-concept only, experiments were limited to inflectional variants, however; future research can easily add other types of variants to the lexicon without changing approach (although of course this increases challenges that may exist with large numbers of tags to be scored due to large numbers of candidate concept combinations cause by combinatorial explosion).

Given the large number of candidate concepts that can result (approximately 1,000 or more for training sentences) the number of potential tags that can result (i.e., different combinations of candidate concepts tagged for different sentence words) can be very large. Only semantically meaningful words in the sentence are tagged with semantically meaningful words in the concept (i.e., 'stop' words like 'the', 'or', 'and', 'with', etc. are ignored).

A map between each word in the candidate concept and matching word in the sentence is maintained but only for the purposes of bookkeeping to calculate the linguistic variables in the terminal set. This map is not used directly in the GP function tree (see TABLE 1).

A. GP Function Tree Evaluation and Training Data

The GP process for evaluating each evolved function tree mimics the process used for tagging a sentence, similar to the MetaMap process where the candidate concept with the highest score is the best 'tag' for a noun phrase. However for this research the tag is a *set* of concepts that applies at the sentence level. Each tag is scored, and the tag with the highest score provides the best set of concepts for that sentence.

Candidate concepts were extracted for a set of five sentences taken from a NLM citation as follows:

> The cases of three patients with a recent history of paralytic poliomyelitis in childhood who developed the flatback syndrome before or after spinal fusion for degenerative disease as adults were reviewed. The flatback syndrome, a combination of an inability to stand erect because of forward flexion of the trunk and pain in the low back and/or legs, typically occurs in the setting of decreased lumbar lordosis as a result of distraction instrumentation of the spine for scoliosis, vertebral fracture, or degenerative disease. Focus was placed on determining the factors responsible for the development and/or persistence of the flatback syndrome in these patients despite maintenance of, or partial operative restoration of, lumbar lordosis. Considering the essential role that the trunk extensor musculature plays in maintaining upright posture, it may be that a new onset of weakness (postpolio syndrome) in this musculature represents a major contributing factor to the flatback syndrome in these patients. Spine surgeons considering operative procedures in patients with a remote history of paralytic poliomyelitis should be aware of the possible increased risk of the flatback syndrome in this population of patients.

FIGURE 5: TEST SENTENCES FROM NLM CITATION – these are tagged with SNOMED-CT concepts for training.

The MetaMap program was run for each sentence and the candidate concepts evaluated. The correct concept tags were saved to an xml file that mapped each concept to the matching word(s) in the sentence. This was used as the training data for the GP evaluation process. Since this is a proof-of-concept only, for simplicity only concepts from the SNOMED-CT ontology were used (this helped accelerate covering space calculations by avoiding uncertainties with whether or not UMLS relationships from other ontologies meet certain conditions necessary for covering space calculations).

Training tags consisted of the correct tag as described above and then a random set of incorrect tags generated from combinations of candidate concepts. Due to the very large number of potential tags (i.e., candidate concept combinations) that could result from each sentence, the tags used for training was limited to approximately 250.

The fitness value used to evaluate each individual in the population is the rank of the correct tag in the list of tags returned by the function tree (ordered in decreasing value) totaled across all sentences.

$$fitness\ f = \sum_{i=1}^{5} rank_{sentence_i} \tag{9}$$

The $rank_{sentence_i}$ is the rank of the correct tag for $sentence_i$ using a zero position list, that is, $rank_1 = 0$.

This is done because the fitness is zero-based, that is, when the correct tag is ranked first then its fitness for that sentence is zero, and consequently if the correct tag is ranked first for all sentences then the total fitness is zero (i.e., Koza style).

B. GP Grammar – Linguistic Variables in Terminal Set

Terminals chosen for the GP grammar that are based upon the MetaMap linguistic calculations include the following:

TABLE 1: LINGUISTIC TERMINAL SET – linguistic variable derived from MetaMap linguistics approach for use in GP approach. Each variable has a value for each tag (i.e., each combination of candidate concepts)

Variable	**Definition**	**Required**
Sentence semantic coverage	Total number of semantically meaningful words in sentence that match words in candidate concepts.	Yes
Number of concepts in tag	Size of the tag set, i.e., number of candidate concepts in tag.	No
Total concept semantic match	Total number of linguistically meaningful words in the candidate concepts that match a word in the sentence.	No
Total concept semantic length	Total number of linguistically meaningful words in all candidate concepts, regardless of whether a match exists in the sentence or not.	Yes
Total concept semantic gap	Total number of linguistically meaningful words in candidate concept not found in sentence.	No
Concept semantic match fraction	Fraction of semantically meaningful words matched in sentence, i.e., total concept semantic match divided by total semantic length.	Yes
Covering space	Total number of concepts in the ontology for the candidate concepts and all ancestors.	No

‘Required’ Variables: In early GP iterations none of the linguistic variables were required, but results were of a form that did not appear to demonstrate the ability to derive fitness functions of a generic enough for to be used for any sentence other than the test sentences. The functions appeared to happen to fit the test sentences by random chance rather than demonstrate the validity of an approach (for example, one GP run resulted in a simple constant divided by the covering space, which has no logical meaning insofar as tagging is concerned). To address this, a minimum number of linguistic variables were identified, as indicated by the ‘Required’ column of Table 1.

‘Covering space’: In the prototype of the semantic search engine the covering space size and intersections between the covering space sets for the documents being searched and the search criteria proved very useful for improving search precision. For this reason the covering space variable was included, but not required, in the GP function tree.

C. GP Software Library

GP evolutions were performed using ECJ Java library for evolutionary computing from George Mason University [23] with development using the Eclipse IDE and Java version 7.

D. GP Configuration and Test Runs

A fixed random number seed was used for each GP run to enable replication.

TABLE 2: BASIC GP CONFIGURATION PARAMETERS – basic Koza-style GP parameter defaults used as indicated in ECJ documentation

Item	**Value**
Maximum # generations	5000
Population size	1024
Maximum tree size	GP run for tree sizes between 4 to 12 in increments of 2 (i.e., 4, 6, 8, 10 & 12)
Tree initialization	Ramp half-and-half (see section 2.2 of [24])
Elitism	None
Crossover probability	90%
Reproduction probability	10%
Random seed	Fixed for each GP thread (two breeding threads and four evaluation threads used)

A GP run was executed for each of the maximum tree sizes listed in Table 2, that is, there were five test runs, one for each of the five tree sizes. Due to the numerous parameters available with ECJ not all are shown.

IV. RESULTS

In all cases a solution was found that ranked the correct tag first for all sentences, and in all cases, within the first few generations (while the GP runs were configured for up to 5000 generations, at most three were required).

Significant bloating occurred when the maximum tree size parameter was set above roughly 8 or higher. These results where are not presented – the purpose of the research is to investigate the feasibility of the method. Future research will investigate refinement of the approach including bloat control.

For GP runs configured with smaller tree sizes the objective functions f that result are as follows (function tree modified to a mathematical format suitable for presentation).

A. GP Results for Max Tree Size = 4

$$f = {(e^{a} - b)}/{e^{c}} \qquad (10)$$

$where$

$$a = conceptCoverageFraction + sentenceCoverage \qquad (11)$$

$$conceptCoverageFraction = \frac{conceptCoverage}{conceptSemanticLength} \qquad (12)$$

$$b = {conceptSemanticLength}/{k} \qquad (13)$$

$$k = 6 \qquad (14)$$

$$c = \frac{conceptSemanticGap}{conceptSemanticLength} \qquad (15)$$

The concept coverage, sentence coverage, concept semantic length, and concept coverage gap variables are defined in Table 1.

Analysis of the function form in (10) suggests that this can be a reasonable approach for ranking and selecting tags. An accurate tag will result from smaller denominator values when the semantic gap for the concepts is minimized by a well fitting tag. This causes the fraction c per (15) to be minimized and hence the denominator to get smaller, thereby of course causing a larger function value. A superior tag will also cause the numerator of the function to increase since sentence coverage will grow, thereby increasing the value of a per (11) and of course the exponential value in the numerator of equation (10), i.e, the value of $(e^{a} - b)$, will increase significantly.

Note the absence of the covering space size. While this variable provides information that enhances search accuracy, it apparently has not provided information that aids in tagging accuracy, at least insofar at the small number of sentences included in this research. Of course future research, when additional sentences are included, may find different results. For example, covering space may have a relationship when looking at all concepts in the document, i.e., when taking into account all prior tags in the document. This can become an approach similar to confabulation theory (per sentence completion experiments in [25]) and hence mimic cognitive brain processes related to language and reading.

B. GP Results for Max Tree Size = 6, 8, 10 and 12

Results for tree sizes 6, 8, 10 and 12, due to bloat, is omitted for brevity.

V. COMMENTS, DISCUSSION, FUTURE WORK

While these results are preliminary, and the intent is to demonstrate a proof-of-concept only, the results appear positive. An objective function was developed using GP for a set of tagged sentences with 100% tagging accuracy.

While the size and diversity of the sentences used for training is limited, the GP process did produce a reasonable solution. The variety of edge conditions will naturally be limited due to the nature of the sampling, but on a preliminary basis these results indicate that an opportunity may exist for sentence tagging using a relatively simple mathematical function.

However, these results are obviously limited for a number of reasons.

- The number of training sentences is limited
- The diversity of the linguistic patterns in the training sentences is limited
- Bloat became problematic as the size of the function tree grew

Of these three limitations, the first two would appear to be approachable via increased sample size and design of the training set. For example, the sampling is purely random but a more effective approach would be to identify a priori patterns in natural language that need to be addressed and ensure these are included in the training data.

In regards to the third point, bloat, current plans are for future research to investigate the use of GramART [26]. GramART applies Adaptive Resonance Theory through the use of BNF grammars. The purpose for using this approach is to leverage the plasticity and stability balance provided by ART-based neural netword designs. As new patterns are identified (in this case changes to the function tree) then the ART plasticity aspect of the neural network adds that pattern to the network, equivalent to adding to the GP function tree. However, if an existing pattern is found it is ignored, which represents the stability benefit. If applied correctly the intended benefit is to remove or reduce the bloat. The form that a tagging objective function will be realized is not yet determined – it may be in the form of a neural network or an interpretable math function.

However, the form of the solution at this point is less important than achieving the overarching goal of simplifying the tagging process when a Bag of Concepts approach is sufficient.

REFERENCES

[1] R. Bai and J. Liao, "Improving Documents Classification with Semantic Features," in Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on, 2009, pp. 640-643.

[2] N. L. o. Medicine. (2013). Unified Medical Language System (UMLS). Available: http://www.nlm.nih.gov/research/umls/quickstart.html

[3] N. L. o. Medicine, "UMLS Terminology Services (UTS)," 2013.

[4] A. R. Aronson, "The effect of textual variation on concept based information retrieval," Proceedings : a conference of the American Medical Informatics Association / ... AMIA Annual Fall Symposium. AMIA Fall Symposium, pp. 373-377, // 1996.

[5] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," Proceedings / AMIA . Annual Symposium. AMIA Symposium, pp. 17-21, // 2001.

[6] A. R. Aronson and F. M. Lang, "An overview of MetaMap: Historical perspective and recent advances," Journal of the American Medical Informatics Association, vol. 17, pp. 229-236, // 2010.

[7] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American Magazine, 2001.

[8] A. Maedche and S. Staab, "Ontology learning for the Semantic Web," Intelligent Systems, IEEE, vol. 16, pp. 72-79, 2001.

[9] H. Davalcu, V. Srinivas, N. Saravanakumar, and I. V. Ramakrishnan, "OntoMiner: bootstrapping and populating ontologies from domain-specific Web sites," Intelligent Systems, IEEE, vol. 18, pp. 24-33, 2003.

[10] H. Haibo, Z. Lei, and Y. Chunxiao, "Semantic-based requirements analysis and verification," in Electronics and Information Engineering (ICEIE), 2010 International Conference On, 2010, pp. V1-241-V1-246.

[11] N. Innab, A. Kayed, and A. S. M. Sajeev, "An ontology for software requirements modelling," in Information Science and Technology (ICIST), 2012 International Conference on, 2012, pp. 485-490.

[12] H. Kaiya and M. Saeki, "Using Domain Ontology as Domain Knowledge for Requirements Elicitation," in Requirements Engineering, 14th IEEE International Conference, 2006, pp. 189-198.

[13] H. Kaiya and M. Saeki, "Ontology based requirements analysis: lightweight semantic processing approach," in Quality Software, 2005. (QSIC 2005). Fifth International Conference on, 2005, pp. 223-230.

[14] M. Kossmann, A. Gillies, M. Odeh, and S. Watts, "Ontology-driven requirements engineering with reference to the aerospace industry," in Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the, 2009, pp. 95-103.

[15] P. Kremen and Z. Kouba, "Ontology-Driven Information System Design," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, pp. 334-344, 2012.

[16] P. Ramadour and C. Cauvet, "An Ontology-Based Reuse Approach for Information Systems Engineering," in Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on, 2008, pp. 572-579.

[17] P. Velardi, S. Faralli, and R. Navigli, "OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction," Computational Linguistics, pp. 655-697, 2012.

[18] H. Inay, O. Kyeong-Jin, and J. Geun-Sik, "Ontology-Driven Visualization System for Semantic Search," in Information Science and Applications (ICISA), 2011 International Conference on, 2011, pp. 1-6.

[19] C. Tianying and F. Hongguang, "Learning Navigation Map on Ontology," in Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on, 2009, pp. 169-174.

[20] L. Zhuhadar, O. Nasraoui, and R. Wyatt, "Visual Ontology-Based Information Retrieval System," in Information Visualisation, 2009 13th International Conference, 2009, pp. 419-426.

[21] A. R. Aronson, "MetaMap: Mapping Text to the UMLS Metathesaurus," UMLS White Paper, 2006.

[22] N. L. o. Medicine, "Lexical Tools (UMLS)," 2013.

[23] G. M. University, "ECJ Evolutionary Computing Java Library," 2013.

[24] R. Poli, L. B. William, and N. McPhee, A Field Guide to Genetic Programming. San Francisco, California: Creative Commons - http://lulu.com, 2008.

[25] R. Hecht-Nielsen, Confabulation Theory The Mechanism of Thought. LaJolla, California: Springer, 2007.

[26] R. Mueth, "Meta-Learning Computational Intelligence Architectures," PhD, Computer Engineering, Missouri University of Science and Technology, 2009.

APPENDIX B

INVERSE COGENCY FOR CONCEPT RECOGNITION

The following was submitted to the Neural Networks journal.

Inverse Ontology Cogency for Concept Recognition

George J. Shannon[1], Bryce J. Schumacher, Donald C. Wunsch II, Steven M. Corns

Missouri University of Science and Technology, Rolla, Missouri, USA

***Abstract* — This paper introduces inverse ontology cogency, a novel measure used in selecting the correct mapping between concepts and words/phrases in free-form text, as encountered in the National Library of Medicine. Inverse cogency is derived from confabulation theory, a non-Bayesian-based theory of cognition. Cogency values predict the cognitive outcome that results from accessing the knowledge base in the cerebral cortex. Inverse cogency leverages this characteristic to identify the most plausible concept in the ontology that matches words or phrases in text. This method is applied as distance measures in a multilayer perceptron neural network used to rank-order candidate concepts during the automated concept recognition process and identify the best match. Hand-annotated text from the National Library of Medicine provides the training and test data. When compared to MetaMap the inverse cogency measure was found to improve concept recognition precision by nearly 5% over the best published results. Inverse cogency used in conjunction with a multilayer perceptron provides a new, effective approach for identifying medical concepts in text.**

***Index terms* — confabulation, cogency, ontology, semantics, natural language processing, semantic tag, concept recognition**

1. INTRODUCTION

Based on the confabulation theory of cognition [3], the inverse ontology cogency measure described in this paper provides a new measure for text-concept mapping, an automated process we name concept recognition. We are using concept-based search, a search method shown to improve precision [6]. In our research, concept-based search retrieves text associated with concepts found in ontologies for the domain of interest, and the search criteria consist of concepts rather than keywords.

Ontologies consist of concepts, i.e., mental notions. They also contain relationships between concepts, where a relationship is a concept. In our research we perform concept recognition for medical text using the SNOMED medical ontology, a subset of the Unified Medical Language System from the National Library of Medicine [8].

Concept-based search using ontologies is not possible unless a concept recognizer associates specific text in the corpus being searched with specific concepts in the domain ontologies of interest. While concept-based search is known to improve precision, the precision of concept-based search is only as good as the precision of the concept recognizer.

We investigated the use of confabulation theory as a way to improve the precision of concept recognition. Confabulation theory is described as a "a new model of vertebrate cognition" that identifies the most plausible conclusions instead of those having the "highest probability of being true," a process named cogent confabulation [1]. It is a model based upon the theory that cognition evolved to maximize survivability within the demands of the environment. According to confabulation theory, the cerebral cortex evolved to contain a fast, feedforward knowledge base that reached greedy (winner take all) conclusions based upon the plausibility of the answer. Plausibility is based upon prior experience (e.g., Hebbian learning), not probability of truth. Cogency is a measure using the product of non-Bayesian conditional probabilities to identify the most plausible outcome of the cognitive process, i.e., the outcome maximizing cogency is the most plausible.

[1] Corresponding author. Email address gjscnc@mst.edu

Confabulation theory has an intuitive appeal since it is based upon finding the most plausible cognitive outcome. This matches our objective of concept-based search. Furthermore, confabulation was shown to work well with: sentence completion experiments demonstrated the ability of confabulation to generate logical sentences, without requiring lexicons or grammar. In these experiments, words for sentence completion were selected based upon maximum cogency alone. The frequencies of co-occurring words in English corpus are used to compute the cogency for candidate words based upon prior words in the sentence, and select the most plausible (maximum cogency) word sequence completing the sentence. This resulted in logical, grammatically correct sentences without the use of linguistics or lexicon [1-4]. These capabilities, along with the simplicity of the cogency measure, made confabulation attractive as the theoretical basis for concept recognition. While the problem of automated concept recognition for ontologies has been addressed using other methods, to our knowledge this paper is the first instance of using the theory of cogent confabulation to aid in recognizing concepts in text that are part of a particular domain ontology.

Inverse cogency is a modified form of cogent confabulation. While cogent confabulation finds the most frequently used word combinations (i.e., from English corpus), inverse cogency identifies the least-likely combination of words that matches the name of a concept. The conditional probabilities for inverse cogency are based upon co-occurring frequencies of words in concept names. This results in cogency values that are limited to the lexicon of the ontology.

When performing the concept recognition process to a group of words, a set of candidate concepts is first retrieved from the ontology. Then, the best match is selected using a distance function to rank-order the candidates. Six features were identified that influence the precision of concept recognition:

1. Fraction of maximum possible inverse cogency for the candidate concept that is mapped to text
2. Fraction of maximum possible inverse cogency for the text that is mapped to the candidate concept
3. Fraction of words in candidate concept mapped to text.
4. Fraction of words in text mapped to candidate concept
5. Whether or not the name of the candidate concept is a single word
6. Whether or not the text being analyzed is a single word

A function to compute the ranking distance using these six features was not readily apparent, so a multilayer perceptron was used as a function approximator, with good results.

Evaluation of the inverse-cogency-based approach was performed using precision. Hand-annotated text from the National Library of Medicine (NLM) was obtained for training and test [51]. These data were manually annotated by NLM staff with the correct mapping between phrases and concepts in the medical ontology, and were used for training and testing the multi-layer perceptron. In addition, we compared the precision of our approach against a popular medical concept recognizer, MetaMap [9-13] from the NLM. MetaMap uses a linguistics-based measure for ranking candidates. Our inverse-cogency-based approach achieved superior performance in comparison with the MetaMap tool.

The content of this paper is as follows:

- Purpose and theoretical background of the inverse cogency measure.
- Definition of the inverse cogency measure.
- Experimental approach and results when using inverse cogency for scoring candidate concepts as part of the multi-layer perceptron.
- Discussion of practical aspects of concept recognition.
- Discussion of the longer-term potential of inverse cogency and confabulation related to search and learning.

2. Overview and Background

2.1 Concept Recognition

We used MetaMap as our baseline for comparison. MetaMap uses noun phrases as the basis for grouping words and performing concept recognition. Hence, a linguistic analysis is performed for each sentence, and then part-of-speech tags are applied to phrases and words, which identify noun phrases for concept recognition. MetaMap retrieves candidate concepts for each noun phrase, with these candidates being extracted from the NLM's Unified Medical Language System. The Unified Medical Language System is a large, public-domain database aggregating multiple medical ontologies [8]. In Step 4 MetaMap scores each candidate using a linguistic heuristic based upon centrality, variation, coverage, and cohesiveness [12].

2.2 Cognitive Relations, Recognition Fidelity, and Recognition Combinatoric Challenges

Word grouping for concept recognition is not constrained to the MetaMap approach of using noun phrases. Cognitive relationships between noun phrases [49] can impact both the precision and fidelity of concept recognition. Concept recognition precision refers to whether or not the words are mapped to the correct concept. We define recognition fidelity as the level of abstractness of the concept mapped, that is, the less abstract, the greater the fidelity.

More specifically, recognition fidelity is inversely related to the distance between a concept and the closest leaf in the ontology. A leaf concept is defined as a concept that has no children. The closer a concept exists to a leaf of the ontology, the less abstract it becomes and the higher its fidelity.

$$\varphi \propto {}^{1}\!\Big/\!\left(1 + d\left(concept_x, concept_{leaf}\right)\right)$$

$where$

φ is fidelity, and,

$d\left(concept_x, concept_{leaf}\right)$ is distance, equal to the length of the shortest path between concept x and the closest ontology leaf

Fidelity is maximized when the concept is a leaf of the ontology graph.

Take, for example, the concept *dorsolumbar spinal fusion with Harrington rod*. This concept is a leaf in the SNOMED ontology. SNOMED is one of the medical ontologies included in the NLM's Unified Medical Language System. An example of using this concept in a sentence is shown in Figure 1 below, which displays the sentence after part-of-speech tagging.

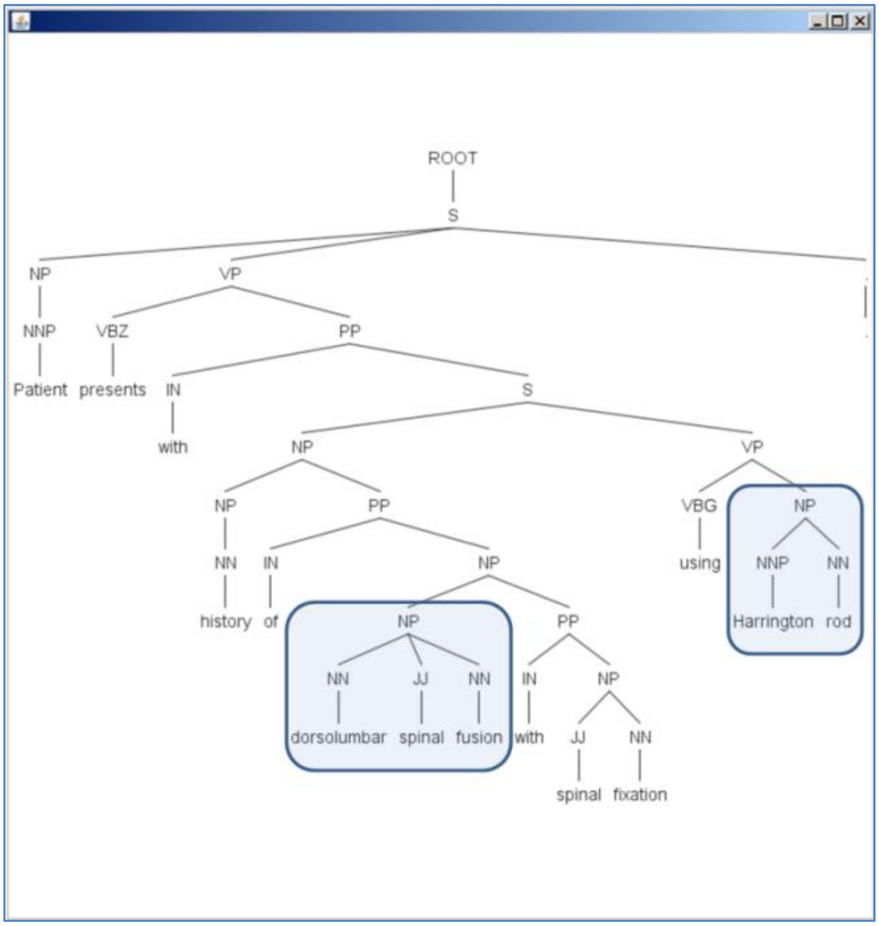


Figure 1: CONCEPT-TEXT TAGGING AT THE PHRASE LEVEL – In this example the concept *dorsolumbar spinal fusion with Harrington rod* is split between noun phrases, causing a reduction in concept recognition fidelity. Future research will investigate alternative approaches, such as the inclusion of cognitive relations found in the sentence to combine related concepts across noun-phrases and thereby recognize higher fidelity concepts.

The sentence splits the concept *dorsolumbar spinal fusion with Harrington rod* across two related phrases. The linkage between the two phrases consists of a conceptual relation from two sentence elements: 1) prepositional phrase "with spinal fixation," and, 2) verb "using." Figure 2 shows the cognitive context of the concept *dorsolumbar spinal fusion with Harrington rod* in the SNOMED ontology. *Harrington rod* also exists as a separate concept, and has a "uses" ontological relationship with *dorsolumbar spinal fusion with Harrington rod* (not shown in Figure 2 since the NLM Semantic Navigator [14] used to obtain these snippets does not provide that level of detail). Since the ontological context matches the conceptual relationship found in the sentence, the two phrases "dorsolumbar spinal fusion" and "Harrington rod" can be combined and mapped to the single concept *dorsolumbar spinal fusion with Harrington rod*. This maximizes fidelity since *dorsolumbar spinal fusion with Harrington rod* is a leaf in the ontology.

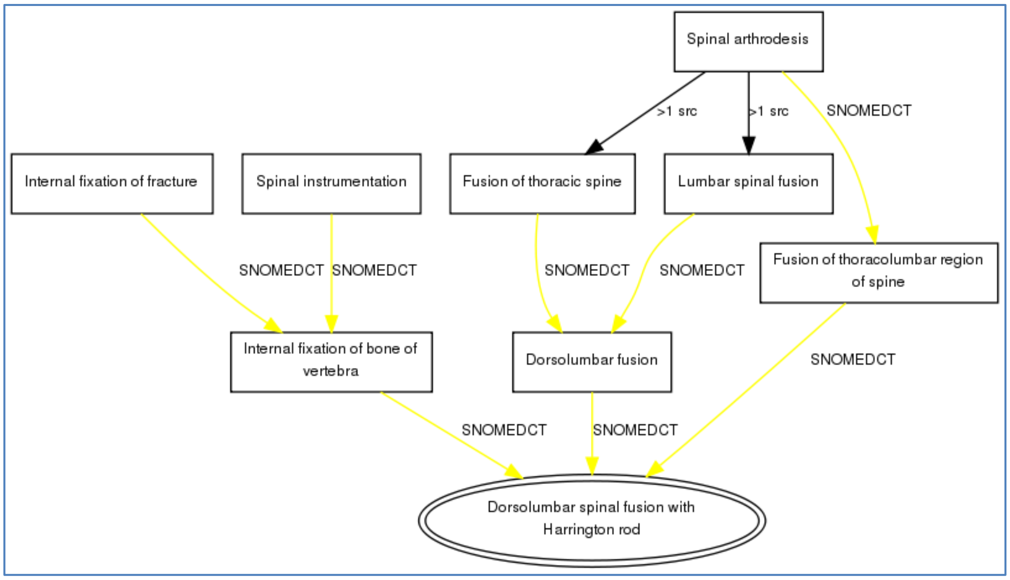
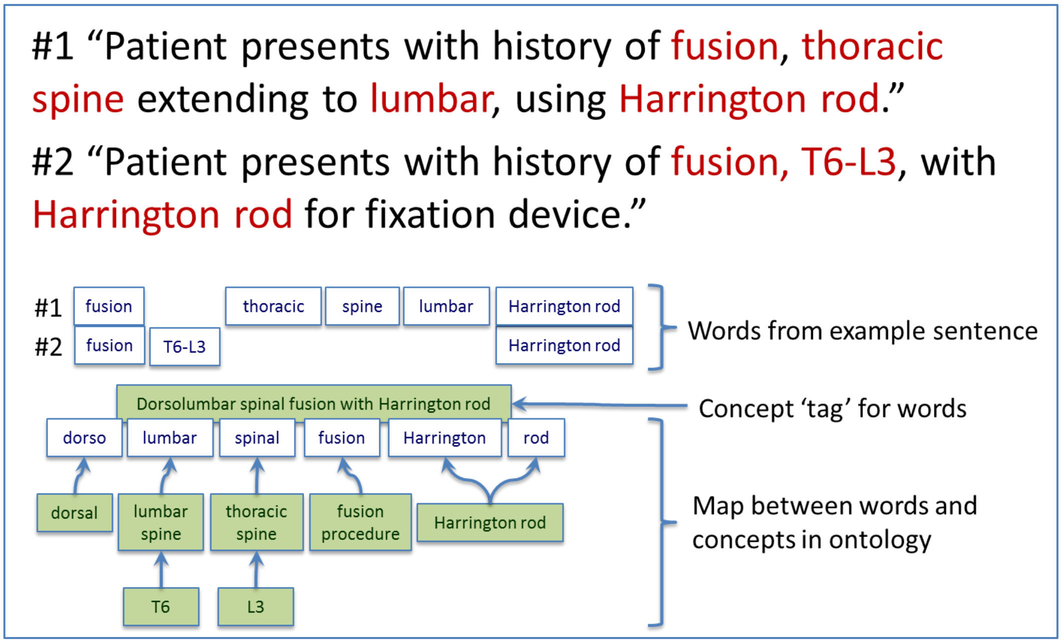


Figure 2: SPINAL FUSION EXAMPLE ONTOLOGY SNIPPET – ontology snippet shows cognitive context of the concept *dorsolumbar spinal fusion with Harrington rod*. A more detailed view of the ontology subsumptive hierarchy will show a separate concept *Harrington rod* that has a "uses" relationship with concept *dorsolumbar spinal fusion with Harrington rod* (details not provided due to space limitations). Ontology snippet from Semantic Navigator, NLM Terminology Services [14].

In real-world text, the cross-phrase cognitive relations are typically more sophisticated than that shown in Figure 1. Figure 3 provides two examples where the concept recognition process used in human cognition implicitly makes use of cognitive relationships in the sentence in combination with relationships in the ontology. For example, sentence 2 of Figure 3 implies anatomic location based upon ontological relationships between the concepts *T6* and *L3* (both vertebrae identifiers) and their respective more abstract anatomic locations *thoracic* and *lumbar*. In this example, although counterintuitive, fidelity is improved by using the more abstract versions of the anatomic concepts *thoracic* and *lumbar* in combination with the procedure and device concepts *fusion* and *Harrington rod* respectively. This example demonstrates how the human cognitive process moves up and down the ontology subsumptive hierarchy, while taking into account cognitive relationships in the sentence, until finding the combination of concepts with the highest fidelity possible.



#1 “Patient presents with history of fusion, thoracic spine extending to lumbar, using Harrington rod.”

#2 “Patient presents with history of fusion, T6-L3, with Harrington rod for fixation device.”

Figure 3: TEXT-CONCEPT TAGGING EXAMPLE – Two examples of tagging sentence text. Both refer to the concept *dorsolumbar spinal fusion with Harrington rod* from the SNOMED medical terminology. This example demonstrates how a variety of facts can be used to identify the best match. A domain expert may implicitly use sentence relationships to infer a concept not explicitly identified by name. An example of this can be found in the second sentence that includes a location-related phrase “T6-L3.” This phrase refers to vertebrae located in the thoracic and lumbar spine regions, respectively. Hence this phrase maps to *dorsolumbar spine* since it refers to specific vertebrae in this same general region.

These examples demonstrate maximizing recognition fidelity via the use of cognitive relations in a sentence and the ontology. However, our research used MetaMap as the baseline for comparing concept recognition precision, and MetaMap performs concept recognition at the noun phrase only. The cognitive relationship between noun phrases is not taken into account.

Furthermore, the number of word map combinations associated with all possible candidate concepts is such that a brute force approach to concept recognition becomes an NP-hard problem. For example, the estimated total number of candidate concepts totals 7.3×10^{15} for an example sentence in the medical domain (see results section for details). Given the difficulty of the problem it is impressive how quickly humans can do this, especially when considering the inclusion of entities and relations in the sentence in combination with entities and relations in the ontology.

Therefore, while we recognize the importance of fidelity in concept recognition, this paper does not take cognition relationships into account. This is done to enable valid comparisons between our approach and MetaMap, which does not account for inter or intra-phrase relationships. Clearly, there is a still-untapped opportunity to use cognitive relationships to improve concept recognition fidelity.

2.3 Confabulation Theory and Inverse Cogency for Concept Recognition

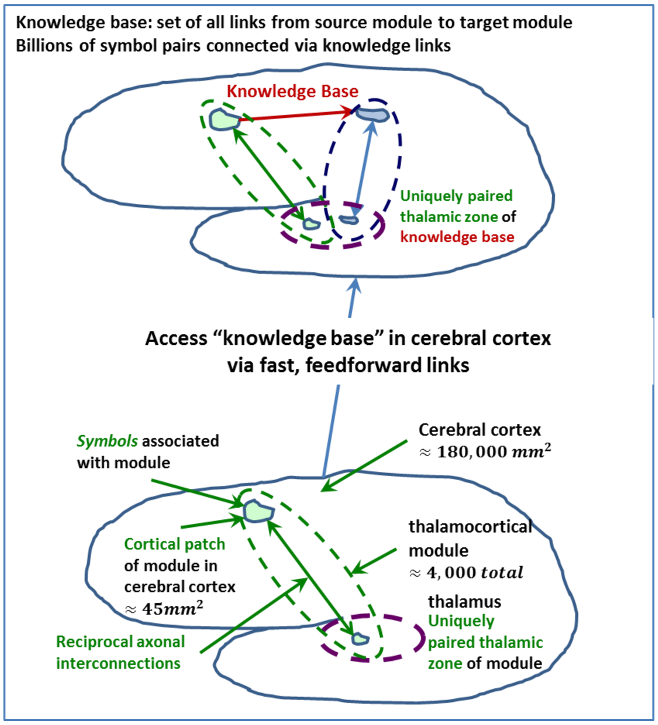

Knowledge base: set of all links from source module to target module
Billions of symbol pairs connected via knowledge links
Knowledge Base
Uniquely paired thalamic zone of knowledge base
Access "knowledge base" in cerebral cortex via fast, feedforward links
Symbols associated with module
Cerebral cortex $\approx 180,000\ mm^2$
thalamocortical module $\approx 4,000\ total$
Cortical patch of module in cerebral cortex $\approx 45mm^2$
thalamus
Uniquely paired thalamic zone of module
Reciprocal axonal interconnections

Figure 4: BRAIN ANATOMY, COGNITION, AND CONFABULATION THEORY – Confabulation theory predicts the outcomes of the fast, greedy, feed-forward neural network architecture composed of the cerebral cortex, thalamus, and cerebral cortex knowledge links which find the most plausible conclusion or action. Adapted from [1, 2].

Confabulation theory [1-4] is based upon evidence that cognitive processing exists via cooperation between approximately 4,000 paired zones in the thalamus and the cerebral cortex (summarized in Figure 4:). Zones of neurons in the thalamus and cerebral cortex reflect attributes of a conceptual notion, where an attribute is stored as a set of neurons in a cortical patch (~ 60 neurons). Each set of neurons defines the neural code for a particular attribute. For example, a set of neurons in the patch for color attributes store the neural code for individual colors.

Excitation of neurons in turn fires cascading signals to other groups via neuronal links. The feed-forward neuronal firing continues until the most plausible ending group is fired (i.e., winner takes all). The final neural group in this chain signals an action or conclusion. For example, a group of neurons related to color, another group related to object shape, and other related to size may result in the final group being related to apples. This final group is the most plausible as it is the group with maximum signal levels.

This winner-take-all process in the context of confabulation theory refers to cognitive processes selecting the first conclusion that appears the most likely from among alternatives. The most plausible conclusion is based upon which neuron group receives the highest signal levels.

In simple terms, the brain makes assumptions about observed events. When an event is observed the cognitive process does not assess the probability that the event actually occurred. If perceived then it is assumed to be factual.

Note that confabulation addresses neuronal processes at a macro level. We use this theory in our study because it provides a simple approach to predicting the outcome of the cognitive process. It does not involve computation of detailed neuronal processes such as neural spiking or timing [52, 53].

2.4 Cogent Confabulation

Figure 5 below summarizes the cogent confabulation process. Cogent confabulation [1] defines cogency as a conditional probability whereas for a set of assumed facts $\lambda = \{\alpha, \beta, \gamma, \delta\}$, the most plausible conclusion ε is the one maximizing the probability:

$$\epsilon = argmax\big(p(\alpha\beta\gamma\delta|\varepsilon)\big)$$

When applied to language cognition, $\alpha\beta\gamma\delta$ is a set of words in text (such as those grouped for concept recognition as discussed earlier) and ϵ is the next word that occurs after them in the left-to-right word sequence. The word set $\alpha\beta\gamma\delta$ is referred to as assumed facts because these words were identified as most plausible in prior confabulation steps.

Hecht-Nielsen, *et al.* [2, 4] report results for sentence completion experiments that apply cogent confabulation via maximization of a proxy measure considered to be "approximate proportional" to cogency as follows:

$$p(\alpha\beta\gamma\delta|\varepsilon) \propto \; p(\alpha|\varepsilon)p(\beta|\varepsilon)p(\gamma|\varepsilon)p(\delta|\varepsilon)$$

$$\varepsilon = argmax\big(p(\alpha|\varepsilon)p(\beta|\varepsilon)p(\gamma|\varepsilon)p(\delta|\varepsilon)\big)$$

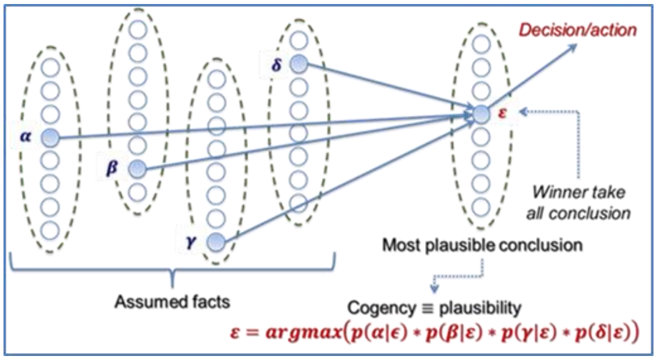

Decision/action
δ
α
β
γ
ε
Winner take all conclusion
Most plausible conclusion
Cogency ≡ plausibility
Assumed facts
$\varepsilon = argmax\big(p(\alpha|\epsilon) * p(\beta|\varepsilon) * p(\gamma|\varepsilon) * p(\delta|\varepsilon)\big)$

Figure 5: CONFABULATION OUTCOME FROM ASSUMED FACTS – The confabulation process simplified consists of a greedy approach based upon the strength of the knowledge link (i.e., cogency). Adapted from [3].

These experiments identified plausibly logical, linguistically correct words for sentence completion without the need for linguistic rules or dictionaries (e.g., no grammar, lexicon, or part-of-speech tags). Furthermore, the experiments demonstrated similar results when the set of assumed facts was extended to include one or more prior sentences.

2.5 Relevance and Relationship between Confabulation, Automated Ontology Learning, and Concept Recognition

Berners-Lee envisioned ontologies as the communication medium for the web, such as their use in the Semantic Web [54, 55]. Ontologies have been used for other purposes such as engineering requirements [16, 17, 20, 21, 24-29]. Ontologies, however, are typically built by hand, and thus are expensive. As a result automated ontology learning has been the subject of much research [38, 40, 56-61].

2.6 Steps Toward Automated Ontology Learning

The process for ontology learning likely involves an iterative learning cycle as follows, of which concept recognition is an important step:

1. Process a logical "chunk' of text, likely a sentence or phrase, and identify new concepts and relationships in text
2. Add these new concepts and relationships to the ontology
3. Perform concept recognition for all text using the new identified concepts
4. Repeat steps 1-3, adding more concepts and relationships, until all text is processed

The ability to automate concept recognition is hence a step towards ontology learning. Moreover, the choice of theoretical framework for concept recognition can influence the approach taken for ontology learning since concept recognition is part of the ontology learning process.

2.7 Confabulation to Aid in Concept Recognition; Ontology as Emergent Property of Confabulation

Our approach to ontology learning focuses on mimicking cognitive processes. This approach is not new; Chen, *et al.* [38, 57] provides examples of automated ontology learning using ART neural networks. In addition, we see entity and relationship recognition as part of the cognitive and ontology learning process. Again, this is not new – it is analogous to cognitive linguistics viewing grammars in terms of "cognitive entities and relations" [50].

What is new in our research is the adaptation of confabulation theory to the problem of concept recognition. Since in our research concept recognition is part of ontology learning, the use of confabulation theory for concept recognition is also part of ontology learning.

What is hypothesized at this time is that the relationship between ontology learning and confabulation is in the interpretation of ontology as a product of confabulation. The ontology can be interpreted as an emergent property of confabulation that represents a portion of the knowledge base stored in the cerebral cortex. This is analogous to viewing grammar and syntax as emergent properties of confabulation theory [2].

The interpretation of ontology as emergent property of confabulation is based upon the correlation between ontology and confabulation as follows:

- An ontological concept correlates to one or more neural codes in the cerebral cortex.
- Ontological relationships and paths correlate to one or more neuronal paths in the knowledge base of the cerebral cortex.

For these reasons the cogency measure appears attractive for adaptation to concept recognition not only for use in concept recognition but also for ontology learning.

2.8 Ontology and Cogency

2.8.1 Reading and Human Concept Recognition in Text

Consider how a possible confabulation cognitive process for concept recognition occurs when reading a sentence, as shown in Figure 6.

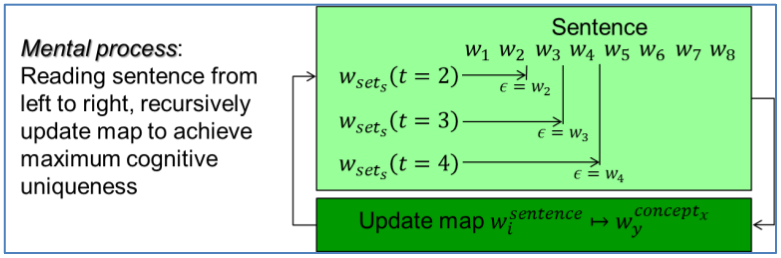

Mental process:
Reading sentence from left to right, recursively update map to achieve maximum cognitive uniqueness
Sentence
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8
$w_{set_s}(t = 2)$
$\epsilon = w_2$
$w_{set_s}(t = 3)$
$\epsilon = w_3$
$w_{set_s}(t = 4)$
$\epsilon = w_4$
Update map $w_i^{sentence} \mapsto w_y^{concept_x}$

Figure 6: CONJECTURE ON RECURRENT MENTAL PROCESS FOR TEXT-CONCEPT MAPPING – As each new word is read, the reader revises their mental map between word groups in the sentence and concepts in the ontology.

When a new word is identified via the symbol-based cognitive process, prior recognized words are re-grouped for optimum concept recognition. This is an iterative approach. Each time a new word is read, combinations of the new word with all prior words can result in a new set of best-fit concepts recognized in the text. This iterative process continues until all words are read.

The sentence reading and concept recognition process takes into account synonyms, punctuation, morphology, acronyms, semantically similar concepts, cognitive relations in the sentence, and so forth. Furthermore, the match between a concept name and phrase is typically not 1:1, yet the confabulation process determines the most plausible match given prior concepts recognized.

2.8.2 Ontology Concept-Name Uniqueness

Concept-text map uniqueness refers to how a reader maps a single concept to a specific set of words in the text. In a word-cerebral cortex mapping study by Huth et al., [62] fMRI scans of subjects listening to the same story demonstrated the following:

- A map exist between an individual word and one or more locations in the cerebral cortex,
- These maps were the same or similar across study subjects,
- If a word has multiple meanings then a different word-location map exists in the cerebral cortex for each meaning, and
- Words tend to map to locations that are co-located by semantic similarity.

In our research we interpreted the Huth, *et al.* word mapping results to be consistent with cogent confabulation. This supports the assumption of a bijective relationship between a group of words, within a particular context, and a unique concept in the ontology. For example, if words have multiple meanings, the confabulation process ensures that the most plausible neural patch is energized, i.e., the neural code associated with the most plausible conceptual interpretation of the word is triggered based upon the context of word use.

Although the Huth, *et al.*, mapping study was for single words only, we assume that a similar process occurs for multi-word concepts while reading text, that is, multiple single-word concepts are energized as each word is recognized in text, and in aggregate this combination is associated with a unique concept. When the location in the cerebral cortex that is associated with a word is energized, the feed-forward paths emanating from this single-word concept are energized. According to confabulation theory, these feed-forward paths are the knowledge links in the knowledge base stored in the cerebral cortex.

The result of energizing each single-word concept, therefore, is energizing paths that are functionally equivalent to energizing one concept associated with all of the single-word concepts. We assume that this occurs such that knowledge links in the cerebral cortex are energized in a manner representing the multi-word concept that exists in the ontology.

To model this, we begin by defining the ontology as a directed acyclic graph as follows:

$Ontology \mapsto DAG(C, R)$,

where C and R are the concept and relationship sets in the ontology, respectively

$C = \{c_1, c_2, \dots c_n\}$, *and,*

$R = \{r_1, r_2, \dots r_m\}$, *where each relationship* r_m *is a concept triple*

$r = \left\{c_{from}, c_{reltype}, c_{to} \middle| c_{from} \in C, c_{reltype} \in C, c_{to} \in C\right\}$

The name $name_c$ for concept c consists of a set of words as follows:

$name_c = \{w_1, w_2, w_3, \dots w_n\}$, *where*

$c \in C$, *and*

$w_1, w_2, w_3, \dots w_n$ *are words in the concept name*

The concept recognition process uses the concept name to determine the optimal map from text to concept, i.e., the map between words in the concept name and the words in the text being analyzed. Avoidance of ambiguous text-concept maps requires the concept name be unique in the ontology, i.e., the concept name is associated with one and only one concept:

$name_c = \{w_1, w_2, w_3, \dots w_n\} \Rightarrow c$,

2.8.3 Cogency and Inverse Cogency Analogy

Cogency for sentence completion experiments [1-3] identifies the most likely next word to follow one or more prior words based upon highest frequency of use. Concept recognition deals with finding one concept that is cognitively unique to the word group, that is, the lowest frequency of use among ontology names.

This is analogous to concept probability. Take, for instance, the thought experiment of randomly selecting a concept from the ontology. Equation (8) shows the probability of any one concept being randomly selected:

$$prob(c) = {}^{1}\!/_{|C|}$$

where

$C = \{c_1, c_2, \dots c_i\}$ *the set of all ontology concepts, and*

$c \in C.$

For example, if the ontology contains 300,000 concepts, the random probability of selecting any one concept is 1:300,000.

The confabulation process for concept recognition must identify a single best-match concept from among many. For example, it selects the single best matching concept out of 300,000. Ideally, one would want it to be implausible for any other concept to be a better match. This analogy is appropriate with the constraint that all concept names are unique in the ontology lexicon and the name maps to a unique concept. Thus cogency relating to concept recognition is referred to as "inverse cogency," where inverse cogency is simply $inverse\ cogency = {}^{1}\!/_{cogency}$. This approach is discussed in more detail in Section 3.

3. Approach

The purpose of this research is to identify a new measure for concept recognition and demonstrate its use and efficacy.

Precision is used as the measure for comparing the efficacy of an approach when ground truth is available, in this case, our approach versus MetaMap[63]:

$$precison = {}^{tp}\!/_{tp + fp},$$

where

$tp = true\ positives, and$

$fp = false\ positives.$

Inverse cogency is used in a MLP to determine whether a candidate concept is an optimum match for a word group.

We also compare inverse-cogency results with results from a random forest [64].

3.1 Sentence Text-Concept Maps

Each word in a word grouping is associated with zero or more candidate concepts. In this paper, a word group is a noun phrase for consistency with the approach used by MetaMap. Candidates for each group are compiled by finding all concepts whose name contains a word in the group. The objective of concept recognition is to select the optimal set of concepts from this aggregate list of candidates.

The artifact that results is a simple concept-word map. Each map provides a 1:1 relationship between a word in the word group and a word in the concept name. Stop words are omitted.

The concept-word map cardinality for word groups, and by association sentence, is shown in **Error! Reference source not found.**:

1. Each word in the sentence has a 1:1 relationship with one word in one concept.
2. Multiple instances of a concept can exist in a word group, but no word can be associated with more than one concept.

The text-concept relationship map is bijective between each word w_i in a word group and each word w_y in the name of a concept, as follows:

$$\mathrm{map}^{\mathrm{concept_x}}: \mathrm{w}_\mathrm{i}^{\mathrm{wgroup}} \mapsto \mathrm{w}_\mathrm{y}^{\mathrm{concept_x}}$$

where

$concept_x \in \mathrm{C}$,

$w_y^{concept_x}$ *is word y in the name of $concept_x$, and*

w_i^{wgroup} *is word i in the word group*

Figure 7 summarizes concept to sentence word cardinality.

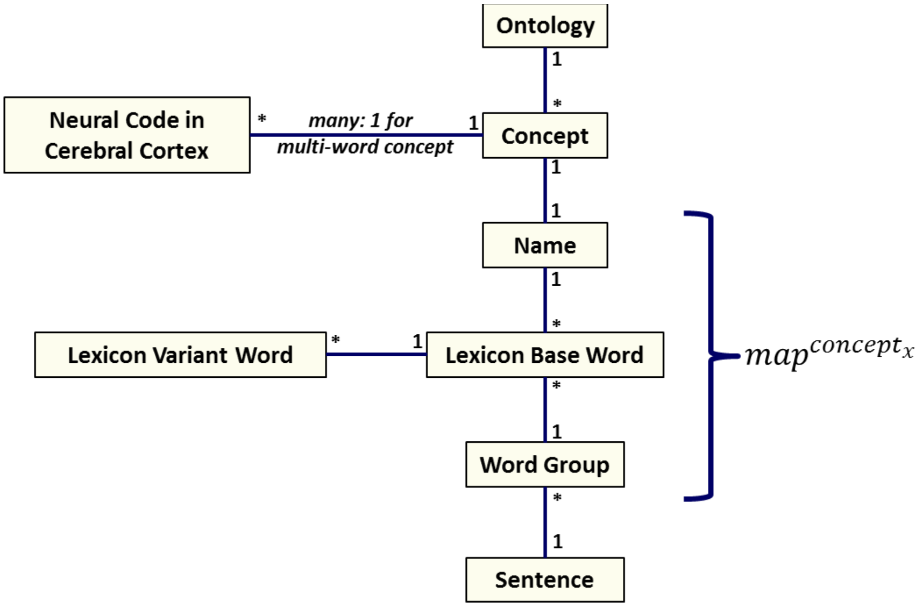


Figure 7: TEXT-CONCEPT MAP OBJECTS – A collection of concept maps for each sentence tag. Here, $map^{concept_x}$ for concept x defines a map between a word in a sentence and a word in a concept name (where the complete map for all concepts in the sentence is $map^{sentence_x} = \{map^{concept_x}\}\forall x$). A neural code in the cerebral cortex, i.e., a patch of neurons, maps to one concept. In our research we assume that if a concept in the domain ontology is a multi-word concept, the concept can be associated with multiple neural codes, based upon results by Huth, *et al.* word mapping experiments.

3.2 Inverse Ontology Cogency

The inverse ontology cogency measure is the inverse of the cogency measure defined in confabulation theory.

The cogency calculation (3) is simply inverted:

$$inverse\ cogency = {}^{1}/_{p(\alpha|\varepsilon)p(\beta|\varepsilon)p(\gamma|\varepsilon)p(\delta|\varepsilon)}$$

The inverse cogency for one concept is the product of the inverted cogency values. The inverse cogency form chosen is logarithmic as follows:

$$IOC(concept_x|w_p) = \begin{cases} -\sum_{i=1}^{n-1} ln\left[p\left(w_i|w_p\right)\right]\ for\ n > 1 \\ 0\ for\ n = 1 \end{cases}$$

where

$IOC(concept_x|w_p)$ *is the inverse ontology cogency for concept x using predicate word* w_p

$n = |N_{concept_x}|$, *where* $N_{concept_x}$ *is the ordered set of words for the concept's name*

$w_i \in N_{concept_x}$ *is the assumed fact word where* $i \neq n$,

$w_p \in N_{concept_x}$ *is the predicate word, and*

$p(w_i|w_p)$ *is the conditional probability of assumed fact word* w_i *and predicate word* w_p *occurring in the same concept name.*

The inverse cogency values are computed for each concept in the terminology. Since any word in the name can be the cogency predicate, an inverse cogency value is computed for each word in the concept name as predicate.

The inverse cogency value for one-word concepts is indeterminate since inverse cogency is not relevant to single-word concepts. To address one-word concepts and word groups, the neural network for scoring candidate concept tags includes inputs indicating whether the concept is a single word and text being tagged is a single word.

4. Testing and Results

4.1 Inverse Cogency Sample

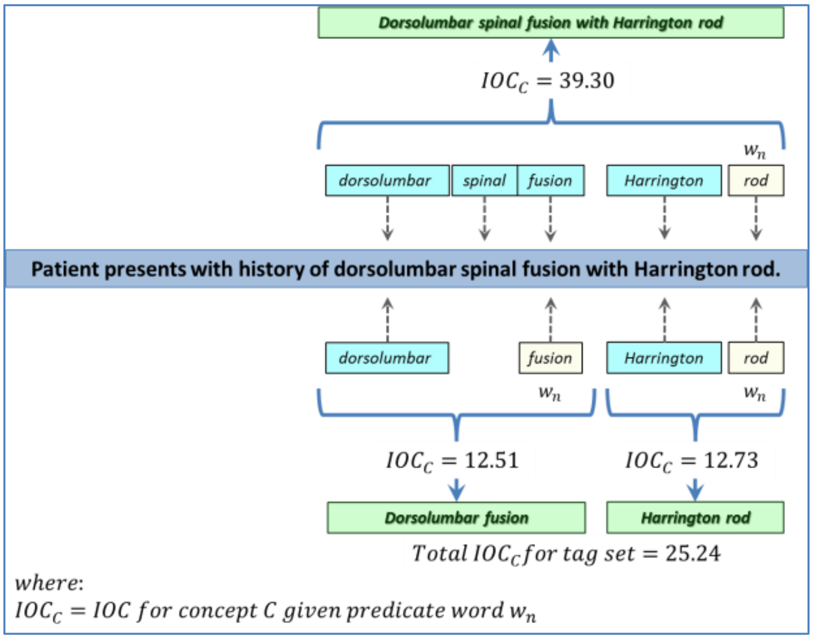

Dorsolumbar spinal fusion with Harrington rod
$IOC_C = 39.30$
w_n
dorsolumbar
spinal
fusion
Harrington
rod
Patient presents with history of dorsolumbar spinal fusion with Harrington rod.
dorsolumbar
fusion
Harrington
rod
w_n
w_n
$IOC_C = 12.51$
$IOC_C = 12.73$
Dorsolumbar fusion
Harrington rod
$Total\ IOC_C for\ tag\ set = 25.24$
where:
$IOC_C = IOC\ for\ concept\ C\ given\ predicate\ word\ w_n$

Figure 8: EXAMPLE OF INVERSE COGENCY FOR CANDIDATE TAGS – The use of the maximum possible inverse cogency for a concept will push the mapping solution towards concepts with multiple words. This example demonstrates how inverse cogency for a concept is maximized for the multi-word concept *dorsolumbar spinal fusion with Harrington rod*.

Figure 8 provides an example of the computing inverse cogency for the concept *dorsolumbar spinal fusion with Harrington rod* compared to an alternative tag set. Inverse cogency favors tags consisting of multi-word concepts over single-word concepts. The alternative tag set is the parent concept *dorsolumbar fusion*, along with the *Harrington rod* tag. Inverse cogency for the correct, higher fidelity tag is greater than the sum of inverse cogency for alternatives. This indicates the potential efficacy of inverse cogency for selecting the optimum tag from candidates.

4.2 Combinatorics, Need for MLP as Universal Approximator

A total of one million sample tag combinations were generated by randomly picking a concept from the list of candidates for each word position in the sentence shown in Figure 8. None of these randomly generated tag sets had a total inverse cogency greater than the inverse cogency for the correct concept tags which provided an early indication that inverse cogency is useful in finding uniquely optimum tags.

To investigate the potential size of the solution space the number of candidate concept combinations was calculated for a hypothetical sentence, as shown in Figure 8. This calculation resulted in 7.3×10^{15} candidates. A straight-forward dynamic programming approach was developed for the entire sentence, as shown in Figure 2. This reduced the number of combinations to 6.8×10^{6}, about nine orders-of-magnitude smaller.

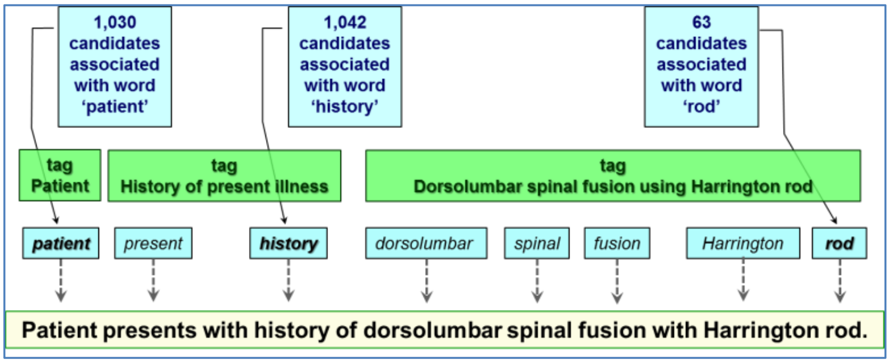

1,030 candidates associated with word 'patient'
1,042 candidates associated with word 'history'
63 candidates associated with word 'rod'
tag Patient
tag History of present illness
tag Dorsolumbar spinal fusion using Harrington rod
patient
present
history
dorsolumbar
spinal
fusion
Harrington
rod
Patient presents with history of dorsolumbar spinal fusion with Harrington rod.

Figure 2: EXAMPLE OF SEMANTIC TAGGING AND COMBINATORIC REDUCTION – In a hypothetical dynamic programming algorithm the analysis of words begins with the last word in the sentence. Matching the words in the sentence to words in each candidate concept will progress from right to left. Starting at the current position in the sentence, and progressing to the left, each word in the candidate is matched to one word in the sentence. When the optimal concept is found, the map between the sentence and concept is frozen for these sentence words. This demonstrates that use of a straight-forward dynamic programming approach has the potential to reduce the solution space significantly.

Solely using dynamic programming, with a simple objective function maximizing the total inverse cogency and words mapped, is insufficient. Approximating the objective function with a neural network was necessary to achieve strong performance. We applied a 6:10:2 MLP.

The input layer consists of the following:

1. Fraction of words in concept name mapped to words in text.
2. Fraction of words in text mapped to words in concept.
3. Fraction of concept maximum possible inverse cogency that is mapped, using Equation 13.
4. Fraction of text maximum possible inverse cogency that is mapped, using Equation 14.
5. Whether or not the text consists of a single word (true = 1, false = 0).
6. Whether or not the concept consists of a single word (true = 1, false = 0).

For $concept_x$, the fraction of mapped inverse cogency versus maximum possible is computed as follows:

$$\rho\left(\mathrm{map}^{\mathrm{concept_x}}, w_p\right) = {IOC\left(\mathrm{map}^{\mathrm{concept_x}}\middle|w_p\right)}\Big/{IOC\left(concept_x\middle|w_p\right)}$$

where

ρ is the fraction mapped concept inverse cogency versus maximum possible,

$\mathrm{map}^{\mathrm{concept_x}}$ *is the bijective map between concept words and the text as shown in Equation (10),*

$w_p \in \mathrm{map}^{\mathrm{concept_x}}$ *is the predicate word,*

$IOC\left(\mathrm{map}^{\mathrm{concept_x}}\middle|w_p\right)$ *is inverse ontology cogency for words in the concept name that are mapped, and*

$IOC\left(concept_x\middle|w_p\right)$ *is the concept inverse cogency calculated using Equation (12).*

For $wgroup_x$, that is, word group x, the fraction of mapped inverse cogency versus maximum possible is computed the same as Equation 10, except that the inverse cogency for the word group is used in the denominator instead of the inverse cogency for the concept name, as follows:

$$\rho\left(wgroup_x, \text{map}^{\text{concept}_\text{x}}, w_p\right) = {IOC\left(\text{map}^{\text{concept}_\text{x}}\middle| w_p\right)}\Big/{IOC\left(wgroup_x \middle| w_p\right)}$$

The output layer consists of two neurons:

1. Score for correct match for this tag (range 0-1).
2. Score for incorrect match for this tag (range 0-1).

The neural network was developed using the Deeplearning4j library [65]. We used backpropagation with a learning rate of 0.01 and a momentum term of 0.09. The hidden nodes used sigmoid activation functions and the output nodes used softmax. A two-node output was chosen where Output 1 is the probability that the candidate is an optimum concept tag and Output 2 is the probability that the candidate is not optimum.

4.3 Training Data: Hand-Annotated Text from NLM

A set of hand-annotated text was made available by the NLM [51]. These were used to train and test the neural network, and to evaluate the precision of MetaMap in selecting the optimum concept. Each annotation provided the correct map between individual words in a phrase and concept for a set of abstracts from the NLM. The annotations were scattered throughout each abstract, that is, all text was not annotated, just a sampling. A summary is in Table 2.

Table 2: COUNT OF NLM HAND-ANNOTATED TEXT

Number of abstracts	592
Number of annotations	3,985

The abstracts were parsed and then stored in a relational database. Parsing of abstracts to extract sentences, phrases, and words was performed using MetaMap to ensure consistency with the manually annotated test data. The parsed words were then matched to the base words in the UMLS lexicon. Matching NLM annotated words to the base form enabled linking the NLM annotations to words in each concept. In some cases new words and word variants were uncovered; these were added to the lexicon.

The NLM annotations provided maps between text and the correct concept down to the word level, which enabled calculating word counts and inverse cogency required for the MLP inputs noted previously.

Training and testing of the neural network was performed using the SNOMED ontology.

NLM manual text-concept annotations typically did not include all words in a phrase. Hence scoring occurred only for the subset of words mapped by the NLM manually annotated data instead of all words in the phrase.

4.4 Neural Network Training and Test Results

4.4.1 Training Data, Data Augmentation, and Training Approach

Training data consisted of the following:

1. NLM manual annotations that map text to concepts in SNOMED
2. For each correct annotation, an incorrect concept tag was drawn randomly from a list of candidates.

In many cases the word set for a concept name was unique in the ontology, and hence a random incorrect concept candidate was not available for reinforcement learning.

Training and validation was performed using a standard 10-k fold with cross-validation. Folds occurred at the concept level, not the annotation level, to ensure random distribution of correct and incorrect concept tags across the folds. Fold assignment for each concept was random.

4.4.2 Neural Network Results

A precision of 80.8% resulted from using the multi-layer perceptron approach. This is 5% better than the best available literature results for MetaMap.

4.4.3 Comparison to Random Forests

The random forest approach was implemented using the R language and the e1071 and caret packages [66, 67]. The random forest approach results in 78.1% precision, which is 2.8% less than the MLP precision result.

4.4.4 Comparison to MetaMap

Literature regarding the precision of MetaMap and Mgrep (a concept tagger from the University of Michigan) indicated a precision for both tools in the 76% range when tagging Medline text [68]. The abstracts in the NLM manually annotated test data are also from the Medline abstract database. This reference study, however, did not use the NLM manual annotations data that we used in our study, so we ran MetaMap against the NLM manual annotations using a local installation on a Windows platform.

This process consisted of executing a MetaMap analysis of each manual annotation (again, using only the annotated portion of a phrase). The concept associated with the maximum MetaMap score was compared to the concept indicated in the data. The results are shown in Table 3.

Table 3: METAMAP PRECISION RESULTS USING NLM HAND-ANNOTATED DATA

Number of Annotations	True Positive	False Positive	Precision
3,644	1,923	1,721	52.8%

This precision is significantly lower than that obtained with our MLP inverse cogency-based approach.

The results in Table 3 come from using the default MetaMap configuration. For example, the default configuration does not include the use of word sense disambiguation. Further optimization of the MetaMap configuration may produce results closer to that found in [68].

4.5 Results Summary

Figure 10 provides a summary of precision results.

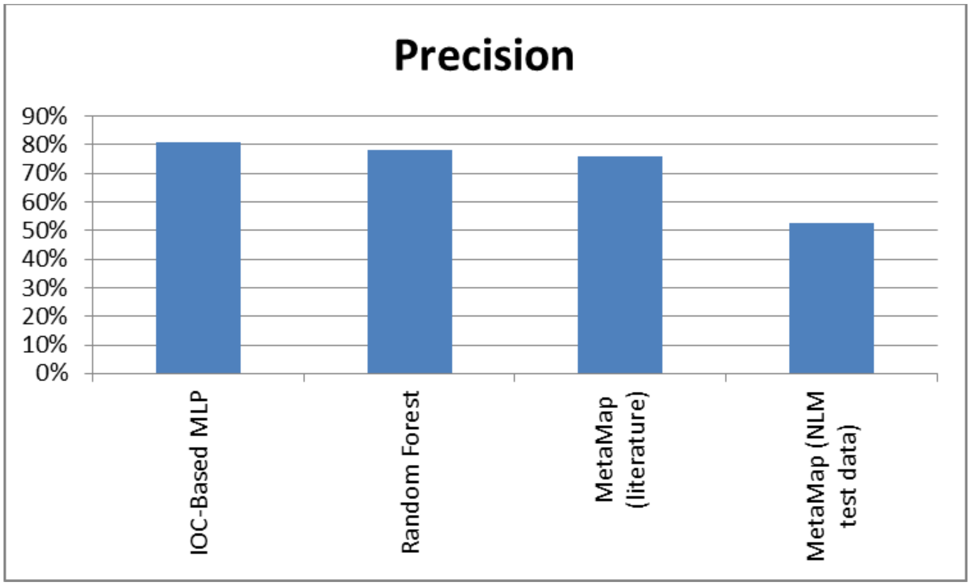


Figure 10: PRECISION RESULTS – the IOC-based multi-layer perceptron has superior performance compared to both MetaMap best in literature, and compared to a local instance of MetaMap performing concept recognition for the manually annotated concept-text maps from the NLM.

5. CONCLUSIONS

The development of the inverse ontology cogency measure offers a new approach for improved precision in concept recognition. This measure is based upon the confabulation theory of cognition that does not rely upon lexicons or grammar. As stated in [2], linguistics such as grammar and syntax "exist only as emergent properties of confabulation." In comparison, MetaMap scores are based upon linguistics. Hence, inverse cogency is significantly different from traditional linguistics-based approaches.

In a comparison with MetaMap, the inverse cogency approach offers superior precision when implemented in a multi-layer perceptron neural network. This advantage can offer a significant improvement in concept-based search precision.

Furthermore, since inverse cogency is based upon a theory of cognition, future work based upon cognitive/biologic mimicry may achieve greater fidelity and further improve precision. Examples include the extraction of cognition relations from sentences as a means to infer cross-phrase concepts for improved search fidelity

References

1. Hecht-Nielsen, R., *Cogent confabulation.* Neural Networks, 2005. **18**(2): p. 111-115.
2. Hecht-Nielsen, R. *The Mechanism of Thought*. in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. 2006.
3. Hecht-Nielsen, R., *Confabulation Theory The Mechanism of Thought*. 2007, LaJolla, California: Springer.
4. Solari, S., et al., *Confabulation Theory*. Physics of Life Reviews, 2008. **5**(2): p. 106-120.
5. Jong-Hwan, K., et al., *Two-Layered Confabulation Architecture for an Artificial Creature's Behavior Selection.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2008. **38**(6): p. 834-840.
6. Moskovitch, R., et al., *A Comparative Evaluation of Full-text, Concept-based, and Context-sensitive Search.* Journal of the American Medical Informatics Association : JAMIA, 2007. **14**(2): p. 164-174.
7. *Ontology - Wikipedia*. Available from: https://en.wikipedia.org/wiki/Ontology.
8. NLM. *Unified Medical Language System (UMLS)*. 2013; Available from: http://www.nlm.nih.gov/research/umls/quickstart.html.
9. Aronson, A., *The Current State of MetaMap and MMTX*. 2009.
10. Aronson, A.R., *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.* Proceedings / AMIA . Annual Symposium. AMIA Symposium, 2001: p. 17-21.
11. Aronson, A.R., *The effect of textual variation on concept based information retrieval.* Proceedings : a conference of the American Medical Informatics Association / ... AMIA Annual Fall Symposium. AMIA Fall Symposium, 1996: p. 373-377.
12. Aronson, A.R., *MetaMap: Mapping Text to the UMLS Metathesaurus.* UMLS White Paper, 2006.
13. Aronson, A.R. and F.M. Lang, *An overview of MetaMap: Historical perspective and recent advances.* Journal of the American Medical Informatics Association, 2010. **17**(3): p. 229-236.
14. NLM. *Semantic Navigator, UMLS Terminology Services (UTS)*. 2013; Available from: https://uts.nlm.nih.gov/home.html.

15. Chan, C.W. *Cognitive informatics: a knowledge engineering perspective*. in *Cognitive Informatics, 2002. Proceedings. First IEEE International Conference on*. 2002.

16. Dang Viet, D. and A. Ohnishi. *Improvement of Quality of Software Requirements with Requirements Ontology*. in *Quality Software, 2009. QSIC '09. 9th International Conference on*. 2009.

17. Haibo, H., Z. Lei, and Y. Chunxiao. *Semantic-based requirements analysis and verification*. in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*. 2010.

18. Huang, S.-L., S.-C. Lin, and Y.-C. Chan, *Investigating effectiveness and user acceptance of semantic social tagging for knowledge sharing.* Information Processing & Management, 2012. **48**(4): p. 599-617.

19. Inay, H., O. Kyeong-Jin, and J. Geun-Sik. *Ontology-Driven Visualization System for Semantic Search*. in *Information Science and Applications (ICISA), 2011 International Conference on*. 2011.

20. Innab, N., A. Kayed, and A.S.M. Sajeev. *An ontology for software requirements modelling*. in *Information Science and Technology (ICIST), 2012 International Conference on*. 2012.

21. Jiehan, Z. and R. Dieng-Kuntz. *Manufacturing ontology analysis and design: towards excellent manufacturing*. in *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*. 2004.

22. Johnson, J., M. Henshaw, and H. Dogan, *An incremental hybridisation of heterogeneous case studies to develop an ontology for capability engineering.* Proceedings of the 22nd Annual International Symposium of the International Council of Systems Engineering, 2012.

23. Kaiya, H. and M. Saeki. *Ontology based requirements analysis: lightweight semantic processing approach*. in *Quality Software, 2005. (QSIC 2005). Fifth International Conference on*. 2005.

24. Kaiya, H. and M. Saeki. *Using Domain Ontology as Domain Knowledge for Requirements Elicitation*. in *Requirements Engineering, 14th IEEE International Conference*. 2006.

25. Kossmann, M., et al. *Ontology-driven Requirements Engineering: Building the OntoREM Meta Model*. in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*. 2008.

26. Kossmann, M., et al. *Ontology-driven requirements engineering with reference to the aerospace industry*. in *Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the*. 2009.

27. Kremen, P. and Z. Kouba, *Ontology-Driven Information System Design.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2012. **42**(3): p. 334-344.

28. Lee, S.W. and R.A. Gandhi. *Ontology-based active requirements engineering framework*. in *Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific*. 2005.

29. Li, S. and L. Shi. *Requirements Engineering Based on Domain Ontology*. in *Information Science and Management Engineering (ISME), 2010 International Conference of*. 2010.
30. Kumar, M., N. Ajmeri, and S. Ghaisas, *Towards knowledge assisted agile requirements evolution*, in *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. 2010, ACM: Cape Town, South Africa. p. 16-20.
31. Merrill, G.H. *A practical multi-ontology approach to knowledge exploration*. in *Biotechnology and Bioinformatics, 2004. Proceedings. Technology for Life: North Carolina Symposium on*. 2004.
32. Ramadour, P. and C. Cauvet. *An Ontology-Based Reuse Approach for Information Systems Engineering*. in *Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on*. 2008.
33. Saad, E.W., et al., *Query-based learning for aerospace applications.* Neural Networks, IEEE Transactions on, 2003. **14**(6): p. 1437-1448.
34. Sarder, B. and S. Ferreira. *Developing Systems Engineering Ontologies*. in *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*. 2007.
35. Soylu, A. and P. De Causmaecker. *Merging model driven and ontology driven system development approaches pervasive computing perspective*. in *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*. 2009.
36. Yun, H. *Research on Building Ocean Domain Ontology*. in *Computer Science and Engineering, 2009. WCSE '09. Second International Workshop on*. 2009.
37. Zhuhadar, L., O. Nasraoui, and R. Wyatt. *Visual Ontology-Based Information Retrieval System*. in *Information Visualisation, 2009 13th International Conference*. 2009.
38. Chen, R.-C. and C.-H. Chuang, *Automating construction of a domain ontology using a projective adaptive resonance theory neural network and Bayesian network.* Expert Systems, 2008. **25**(4): p. 414-430.
39. Hourali, M. and G.A. Montazer, *A New Approach for Automating the Ontology Learning Process Using Fuzzy Theory and ART Neural Network.* Journal of Convergence Information Technology, 2011. **6**(10): p. 24-32.
40. Cross, V. and V. Bathija, *Automatic ontology creation using adaptation.* AI EDAM, 2010. **24**(Special Issue 01): p. 127-141.
41. Davalcu, H., et al., *OntoMiner: bootstrapping and populating ontologies from domain-specific Web sites.* Intelligent Systems, IEEE, 2003. **18**(5): p. 24-33.
42. Dongyeop, K., et al. *Automatically learning robot domain ontology from collective knowledge for home service robots*. in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*. 2009.
43. Navigli, R., P. Velardi, and A. Gangemi, *Ontology learning and its application to automated terminology translation.* Intelligent Systems, IEEE, 2003. **18**(1): p. 22-31.
44. Navigli, R. and P. Velardi. *LearningWord-Class Lattices for Definition and Hypernym Extraction*. in *48th Annual Meeting of the Association for Computational Linguistics*. 2010. Uppsala, Sweden.

45. Velardi, P., S. Faralli, and R. Navigli, *OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction.* Computational Linguistics, 2012: p. 655-697.
46. Liu, C., et al., *Convolution Neural Network for Relation Extraction.* Advanced Data Mining and Applications, 2013. **8347**: p. 231-242.
47. Schenker, A., et al., *Graph-Theoretic Techniques for Web Content Mining*. Series in Machine Perception and Artificial Intelligence, ed. H. Bunke and P.S.P. Wang. 2005, Hackensack, NJ: World Scientific Publishing Co. Pte. Ltd.
48. Croft, W. and D.A. Cruse, *Cognitive Linguistics*. 2004: Cambridge University Press. 356.
49. Radden, G. and R. Dirven, *Cognitive English Grammar*. Cognitive Linguistics in Practice, ed. G. Radden. 2007, Amsterdam, The Netherlands: John Benjamins Publishing Company. 374.
50. Pipitone, A. and R. Pirrone. *Cognitive Linguistics as the Underlying Framework for Semantic Annotation*. in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. 2012.
51. Medicine, N.L.o., *MEDLINE citations annotated with disorder mentions*. 2015.
52. Gerstner, W., et al., *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. 2014, Cambridge, United Kingdom: Cambridge University Press. 577.
53. Bastiaansen, M. and P. Hagoort, *Oscillatory neuronal dynamics during language comprehension*, in *Progress in Brain Research*, K. Christa Neuper and Wolfgang, Editor. 2006, Elsevier. p. 179-196.
54. Berners-Lee, T., *WWW: past, present, and future.* Computer, 1996. **29**(10): p. 69-77.
55. Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web.* Scientific American Magazine, 2001(May).
56. Maedche, A. and S. Staab, *Ontology learning for the Semantic Web.* Intelligent Systems, IEEE, 2001. **16**(2): p. 72-79.
57. Chen, R.-C., J.-Y. Liang, and R.-H. Pan, *Using recursive ART network to construction domain ontology based on term frequency and inverse document frequency.* Expert Systems with Applications, 2008. **34**(1): p. 488-501.
58. Fortuna, B., N. Lavrač, and P. Velardi, *Advancing Topic Ontology Learning through Term Extraction*, in *PRICAI 2008: Trends in Artificial Intelligence*, T.-B. Ho and Z.-H. Zhou, Editors. 2008, Springer Berlin Heidelberg. p. 626-635.
59. Gherasim, T., et al., *Methods and Tools for Automatic Construction of Ontologies from Textual Resources: A Framework for Comparison and Its Application*, in *Advances in Knowledge Discovery and Management*, F. Guillet, et al., Editors. 2013, Springer Berlin Heidelberg. p. 177-201.
60. Healy, M.J. and T.P. Caudell. *Generalized Lattices Express Parallel Distributed Concept Learning*. in *Fuzzy Systems, 2006 IEEE International Conference on*. 2006.
61. Zhang, R.-l. and H.-s. Xu. *Using Bayesian Network and Neural Network Constructing Domain Ontology*. in *Computer Science and Information Engineering, 2009 WRI World Congress on*. 2009.
62. Huth, A.G., et al., *Natural speech reveals the semantic maps that tile human cerebral cortex.* Nature, 2016. **532**(7600): p. 453-458.

63. Manning, C.D. and H. Schütze *Foundations of Statistical Natural Language Processing*. 1999, Cambridge, Massachusetts: The MIT Press. 680.
64. Ho, T.K., *Random Decision Forests*, in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. 1995: Montreal, Canada. p. 278-282.
65. Deeplearning4j, D.T., *Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0*.
66. Meyer, D., et al., *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. 2015.
67. Kuhn, M., et al., *caret: Classification and Regression Training*. 2016.
68. Shah, N.H., et al., *Comparison of concept recognizers for building the Open Biomedical Annotator.* BMC Bioinformatics, 2009. **10**(Suppl 9): p. S14.
69. Organization, I.H.T.S.D., *SNOMED CT*.

APPENDIX C

COGNITIVE RELEVANCE

The following is a journal article for submission to IEEE Transactions on Biomedical Engineering.

Cognitive Relevance

George J. Shannon
Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, Missouri, USA
gjscnc@mst.edu

Steven M. Corns
Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, Missouri, USA
cornss@mst.edu

James M. Levett, MD FACS
Chief Medical Officer
Physician's Clinic of Iowa
Cedar Rapids, Iowa, USA

Donald C. Wunsch II
Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, Missouri, USA
dwunsch@mst.edu

***Abstract* —.This paper discusses the results of investigating simple, cognitive-based approaches to search. The emphasis is placed on simplicity, and determining if a simple ranking measure is sufficient for improved search precision. The measures chosen are concept-based since concept and context-based search improves precision. These results provide direction on the need for more complicated methods. If a simple, yet effective, distance measure is found for rank-ordering search results for improved precision, then approaches may be feasible for improving search precision in a shorter period of time at less cost. Moreover, the methods investigated use a natural language interface that enables far more complicated criteria while remaining intuitive to the casual user. Furthermore, these criteria better reflect search requirements than keywords alone. Two cognitive measures were investigated: a topology-based measure, and a cogency-based measure. The topology-based measure uses a covering space algorithm for the domain ontology, quantifying the size of the intersection of the topological covering space of the search criteria and covering space of the document in the corpora being searched. This covering space, based upon the subsumptive property of the ontology, creates a set of imputed concepts that are cognitively relevant. The cogency-based measure, along with the ontological structure itself, is consistent with the confabulation theory of cognition, serving as a proxy for the knowledge base stored in the cerebral cortex. It is also consistent with cognitive linguistics. The corpus for testing search precision was sampled from NLM publication abstracts, and search results were scored by a physician. Results indicate that improving search precision via the simple use of these two measures, even though related to cognition, are insufficient for improving search precision. While a simple ranking metric is preferred, the results suggest that efforts to improve search precision are better spent on more complicated methods, for example, neural network-based approaches.**

***Index terms* — ontology, search, search relevancy, semantic search, cognitive search, healthcare informatics**

I. INTRODUCTION

Two simple cognitive relevance measures are presented here for rank-ordering search results to improve precision. These are the topology-based measure and the cogency-based measure. The basis for both of these measures is explained in the context of a theory of human cognition [1-3] and cognitive linguistics [4]. Furthermore, benefits from use of a cognitive-based relevance measure in conjunction with a fully natural language user interface are explored.

The topology-based measure uses the ontology covering space for scoring search relevance. It leverages the subsumptive property of ontologies, which is used to determine the cognitive context of both the search criteria and the text being searched.

The approach described for the topology-based measures requires use of a domain ontology and a tool for automated concept recognition in text. The benefit is that this enables a natural language user interface. The ability to create complicated search criteria is an inherent outcome of the natural language interface, and negates the need for complicated search logic. Such an interface provides end-users with the ability to create complicated search criteria in an intuitive manner, thereby masking from the average user the complexity of creating sophisticated search criteria. In practical terms the natural language interface avoids overly simplistic criteria necessary for a simplified user interface, or a complex interface required for complicated search logic. Furthermore, the natural language interface requires little or no training for the casual user.

The cogency-based measure uses the cogency measure described in the cogent confabulation theory of cognition [1-3]. Maximization of the conditional probabilities of co-occurring concepts in text, in theory, identifies the most likely cognitive fit. This is a straight-forward application of the cogency measure, a desirable attribute when striving for simplicity.

Section II introduces the notion of concept-based search, along with its relationship to a cognition theory, cognitive linguistics, and ontologies. It defines the simple ontology covering space measure for quantifying the relevance of search results. This investigates if the ontology structure, as a proxy for the knowledge networks in the human brain, can be used to quantify how close a search result is to the criteria. In particular this section defines how the ontological relationships provide the subsumptive covering space and how the covering space leverages the complicated part/whole aspect of knowledge. The cogency-based measure is also defined in Section III. Section IV presents the testing approach and results. This includes an example of the natural language search criteria used for testing. Section V addresses conclusions.

II. ONTOLOGY, COGNITION, SEMANTIC SEARCH

Ontology, cognition theory, semantic search, and cognitive grammar are reviewed to provide the rationale for use of the ontology structure to compute cognitive relevancy.

A. Ontology

1) Medical Ontologies and Tools Used in Research

The Systematized Nomenclature of Medicine (SNOMED) ontology used in this research is a subset of the Unified Medical Language System (UMLS) [5] available from the National Library of Medicine (NLM). The UMLS version used for our research consists of 2,493,384 concepts. The SNOMED subset used for our research consisted of 323,292 concepts, and was chosen due to the clarity and simplicity of its relationships. These relationships provide the context needed for the covering space calculations (to be addressed later in this paper). The UMLS was needed since it contains certain metadata required for the NLM MetaMap tool (the tool used for automated concept recognition in text). Use of the UMLS for contextual relationships was not feasible due to the ambiguity of the direction and uniqueness of its relationships, hence, research was limited to SNOMED subset due to the concise and understandable relationship structure.

The ontology structure used by SNOMED is quite simple and was stored in two simple relational database tables as shown in Figure 1, and consists of the following:

- Concepts: unique identifier, name, and whether or not the concept is a relationship type
- Relationships: from concept, to concept, and type of relationship. A relationship type is itself a concept.

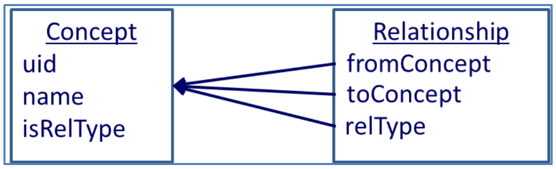


Figure 1: SIMPLE ONTOLOGY PERSISTENCE SCHEMA - the ontology was stored in a simple relational database structure.

SNOMED contains a hierarchy of relationship types but these are not relevant to our purposes.

No predicate logic was used in our research. Instead, the algorithm for computing relevancy used only existing ontology relationships. This minimized algorithmic complexity. If new relationship types were required to identify the cognitive covering space, it may require fairly sophisticated rules logic. An example of logic rules for instantiating new relationships can be found in Kumar, *et al.* [6].

2) *Ontology Subsumption and Features*

Subsumption refers to the ontological property of concepts representing more abstract, broad notions and more specific, narrow notions concepts that it encompasses. A parent subsumes all of its descendent concepts, that is, it refers to the "is a" hierarchical relationship between parent and child. The parent is the more abstract concept in comparison to the child, the child's children, ad infinitum. A more detailed explanation of subsumption along with an example of extracting subsumptive relationships from text can be found in [7].

The subsumptive properties of the ontology results in a number of features that are important to the development of the cognitive relevance measure as follows:

- <u>Relationship Types</u>: The subsumptive relationship type, the "is a" relationship, is required. Relationship types can also include zero or more non-subsumptive relationship types. A relationship type is a concept.

- <u>Directed Relationships and Subsumption Requirement</u>: All relationships emanate "upwards" conceptually from the most specific concept towards the most abstract concept, forming the subsumptive hierarchy. For SNOMED this results in a small set of categorical concepts at the highest conceptual level (procedure, anatomy, device, etc.). If a relationship emanating from a concept is non-subsumptive, the "to" concept that the non-subsumptive relationship points to is part of a one or more subsumptive hierarchies, by definition.

- <u>Multiplicity of Subsumption</u>: The ontology allows multiple subsumptive parents, i.e., a concept may conceptually exist in multiple high-level categories. An example of this is a multi-word concept that encompasses multiple parental concepts, e.g., a multi-word concept may encompass a procedure, anatomical location, and device used.

- <u>Specificity – Abstractness Relationship</u>: The specificity of a concept increases as the distance between it and concepts at the highest abstract level increases.

- <u>Ontological Graph</u>: The ontology can be represented as a directed, acyclic graph. Points in the graph are concepts and connections are instances of a relationship type.

3) *SNOMED Ontology Example*

Take for example a concept in the healthcare domain, as shown in Figure 2.

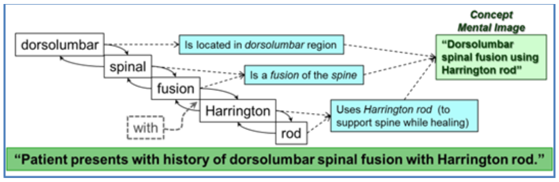

Concept
Mental Image
dorsolumbar
spinal
fusion
with
Harrington
rod
Is located in *dorsolumbar* region
Is a *fusion* of the *spine*
Uses *Harrington rod* (to support spine while healing)
"Dorsolumbar spinal fusion using Harrington rod"
"Patient presents with history of dorsolumbar spinal fusion with Harrington rod."

Figure 2: SNOMED CONCEPT EXAMPLE – Example of a complex, multi-word concept that encompasses multiple conceptual categories.

The concept "dorsolumbar spinal fusion with Harrington rod" shown in Figure 2 is one concept in the SNOMED ontology. This concept is in the anatomy, procedure, and device categories (see Figure 3 below).

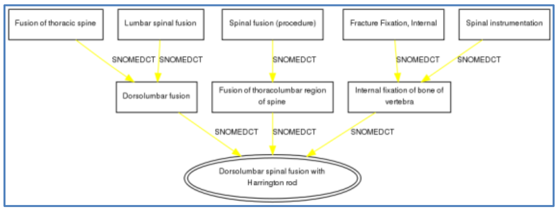

Fusion of thoracic spine
Lumbar spinal fusion
Spinal fusion (procedure)
Fracture Fixation, Internal
Spinal instrumentation
SNOMEDCT
SNOMEDCT
SNOMEDCT
SNOMEDCT
SNOMEDCT
Dorsolumbar fusion
Fusion of thoracolumbar region of spine
Internal fixation of bone of vertebra
SNOMEDCT
SNOMEDCT
SNOMEDCT
Dorsolumbar spinal fusion with Harrington rod

Figure 3: SIMPLIFIED ONTOLOGY SNIPPET - Small subset of subsumptive hierarchy from the concept "dorsolumbar spinal fusion with Harrington rod" (provided by NLM's Terminology Services https://uts.nlm.nih.gov/home.html).

The actual number of concepts in the simplified hierarchy shown in Figure 3 is far more than that shown. In reality there are over 100 related concepts at higher, more abstract cognitive levels. As shown in this example, the subsumptive relationships can imply a large number of more abstract concepts. Obviously this is not taken into account by search engines based upon keywords alone.

B. Theory of Cognition

The ontology structure essentially serves as a proxy for the knowledge base stored in the cerebral cortex.

Confabulation theory [1-3] explains cognition as a process that accesses the neural codes and relationships in the cerebral cortex. This process is instantiated via thalamocortical links between the thalamus and the cerebral cortex.

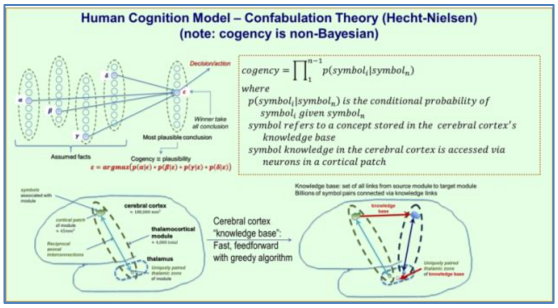

Human Cognition Model – Confabulation Theory (Hecht-Nielsen)
(note: cogency is non-Bayesian)
Decision/action
α
β
γ
δ
ε
Winner take all conclusion
Most plausible conclusion
Assumed facts
Cogency ≡ plausibility
$\varepsilon = argmax(p(\alpha|\varepsilon) * p(\beta|\varepsilon) * p(\gamma|\varepsilon) * p(\delta|\varepsilon))$
$cogency = \prod_{1}^{n-1} p(symbol_i|symbol_n)$
where
$p(symbol_i|symbol_n)$ is the conditional probability of $symbol_i$ given $symbol_n$
symbol refers to a concept stored in the cerebral cortex's knowledge base
symbol knowledge in the cerebral cortex is accessed via neurons in a cortical patch
symbols associated with module
cerebral cortex
cortical patch of module
Reciprocal axonal interconnections
thalamocortical module
≈ 4,000 total
thalamus
Uniquely paired thalamic zone of module
Cerebral cortex "knowledge base": Fast, feedforward with greedy algorithm
Knowledge base: set of all links from source module to target module
Billions of symbol pairs connected via knowledge links
knowledge base
Uniquely paired thalamic zone of knowledge base

Figure 4: BRAIN ANATOMY, COGNITION, AND CONFABULATION THEORY - Confabulation theory predicts the outcomes of the fast, greedy, feed-forward neural network architecture composed of the cerebral cortex, thalamus, and knowledge links which find the most plausible. Adapted from [1, 3].

Confabulation theory is based upon evidence that the human cognitive process exists via cooperation between approximately 4,000 paired zones in the thalamus and the cerebral cortex (summarized in Figure 4). Zones of neurons in the thalamus and cerebral cortex reflect attributes of a conceptual notion, where an attribute is stored as a set of neurons in a cortical patch (typically ~ 60 neurons). Each set of neurons defines the neural code for a particular attribute. For example, a set of neurons in the patch for color attributes store the neural code for individual colors, e.g., blue. Excitation of such a set of neurons fires cascading signals to other groups via knowledge links.

Note that confabulation theory and cogency address neuronal dynamics at a macro level. Neither confabulation theory nor the cogency measure attempts to delve into the details of sophisticated neural processes, e.g., neuron spiking or timing, such as that discussed in [8, 9].

The feed-forward neuronal group firing continues until the most plausible ending group is fired (i.e., winner takes all). The final group in this chain signals an action or conclusion. For example, a group of neurons related to color, another group related to object shape, and other related to size may result in the final group being related to apples. This final group is the most plausible, that is, the group with maximum cogency.

Cogent confabulation [1] defines cogency as a conditional probability whereas for a set of assumed facts $\lambda = \{\alpha, \beta, \gamma, \delta\}$, the most plausible conclusion ε is the one maximizing the probability:

$$\epsilon = argmax\big(p(\alpha\beta\gamma\delta|\varepsilon)\big) \qquad (1)$$

If confabulation is applied to language cognition then $\alpha\beta\gamma\delta$ is a set of words in a phrase or sentence, and ϵ is any word likely found to occur after them in the temporal sequence. The word set $\alpha\beta\gamma\delta$ is referred to as assumed facts because these words were identified in prior confabulation steps. Cogency does not make use of the probability that their perceived existence is accurate, i.e., it is non-Bayesian.

Hecht-Nielsen, *et al.* [3, 10] reported results for sentence completion experiments that apply cogent confabulation via maximization of a proxy measure considered to be "approximate proportional" to cogency as follows:

$$p(\alpha\beta\gamma\delta|\varepsilon) \propto\ p(\alpha|\varepsilon)p(\beta|\varepsilon)p(\gamma|\varepsilon)p(\delta|\varepsilon) \qquad □2□$$

$$\varepsilon = argmax\big(p(\alpha|\varepsilon)p(\beta|\varepsilon)p(\gamma|\varepsilon)p(\delta|\varepsilon)\big) \qquad (3)$$

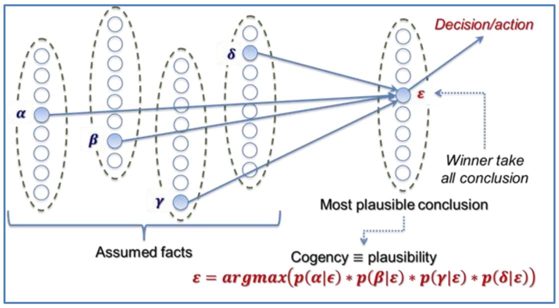

Decision/action
α
β
γ
δ
ε
Winner take all conclusion
Most plausible conclusion
Assumed facts
Cogency ≡ plausibility
$\varepsilon = argmax(p(\alpha|\epsilon) * p(\beta|\varepsilon) * p(\gamma|\varepsilon) * p(\delta|\varepsilon))$

Figure 5: CONFABULATION OUTCOME FROM ASSUMED FACTS – The confabulation process simplified consists of a greedy approach based upon the strength of the knowledge link (i.e., cogency). Adapted from [3].

Notable about these experiments was the identification of plausibly logical, linguistically correct words to complete a sentence without the need for either linguistic rules or dictionaries (e.g., grammars, lexicons, or part-of-speech tags). Furthermore, these experiments demonstrated similar results when the set of assumed facts was extended to include prior sentences. The conclusion is that grammar and syntax "exist only as emergent properties of confabulation" [10].

C. *Semantic Brain Map and the Ontology*

In Huth, *et al.* [11], fMRI imaging, taken while test subjects listen to scripted stories, produced maps, called "semantic tiles," of the physical location of concepts stored in the cerebral cortex. Patterns of similar storage locations for the same concepts were observed across study subjects. It also provided evidence of the physical co-location of similar concepts, i.e., semantic grouping. Conversely, it demonstrated that a different meaning of the same word is stored in a different location, i.e., the cerebral cortex stores concepts, not words.

This research is consistent with the confabulation viewpoint that the cerebral cortex stores a person's knowledge as a network of interconnected concepts. An interpretation of this consistency, as it relates to the use of ontological data to determine cognitive relevance, is that the ontology is an emergent property of cognition. This is simply an extension of Hecht-Nielsen's interpretation of linguistics as "an emergent property of confabulation" [10].

In the research discussed in this paper, the ontology, as a written and graphical artifact of cognition, is interpreted to have a structure that reflects the cognitive structure of these concepts in the cerebral cortex. This structure consists of a unique neural patch/code for each unique concept, links between related concepts, along with the physical co-location of related concepts. It is assumed that physical co-location occurs to reduce latency in identifying related concepts.

Ontology subsumption, along with confabulation theory, semantic tiling, and interpreting the ontology as an emergent property of cognition, provided the theoretical basis for the development of the cognitive relevance measure.

D. Semantic Search

Semantic search, also called concept-based search, refers to the search method of finding mental notions in lieu of keywords. Concept-based search looks for a specific concept rather than a list of keywords. For example, when the search criteria consist of "dorsolumbar spinal fusion with Harrington rod", a concept-based search has only to look for one concept. But a keyword search must look for all keywords and the possible combinations of these.

From this perspective the shortcoming of keyword search is that they can be grouped into multiple different combinations that infer fundamentally different cognitive notions, thereby biasing search results. Concept-based search, along with context-based search, however, can improve precision [12]. The approach provided in this paper is a step towards blending concept and context search.

This does not mitigate the difficulties of natural language processing. For example, developing an automated method that maps text to concepts in the ontology can be challenging.

In our research we used MetaMap [13-18] from the NLM to perform this mapping. MetaMap made it possible to "tag" words and phrases in the natural language search criteria with the matching concepts. The same was performed for corpora text being searched. As described later in this paper, these tags are used to extract an ontological covering space from both the search criteria and corpora text and determining the intersection of the two. This intersection is the basis for quantifying the relevancy of search results.

E. Cognitive Linguistics

Cognitive linguistics is based upon the premise that "language is governed by general cognitive principles, rather than by a special-purpose language module" [19]. Linguistic operations relate to general cognitive processes. The three major hypotheses for cognitive linguistics are (as stated in [19]), as follows:

1. Language is not an autonomous cognitive facility
2. Grammar is conceptualization
3. Knowledge of language emerges from language use

Cognitive linguistics appears consistent with the cognition theory per Hecht-Nielsen as shown in Table 1 below.

Table 1: COGNITIVE LINGUISTICS RELATION TO COGNITION THEORY

Cognitive Linguistics	**Cognition Theory (Confabulation)**
Language is not autonomous facility	Knowledge base stored in the greedy feedforward networks in cerebral cortex is used for all cognition, including language
Grammar is conceptualization	Neural patches contain the neural codes for attributes and conceptual notions (neural code exists for words, and via the feedforward networks, link to other patches representing cognitive notions in the knowledge base)
Language emerges from language use	Language experiments using sentence completion, based solely upon conditional probabilities computed from prior language use, produced rational and linguistically correct sentences without the use of lexicon, grammatical analysis, or linguistic rules

An example in medicine is the linguistic construal of topological or geometric structure that is represented in the ontology. Cognition regarding the concept "dorsolumbar spinal fusion with Harrington rod" is shown in the ontology snippet in Figure 3, demonstrating the ontological equivalence of "construal" of anatomic location via subsumptive relationships. Other concepts are easily construed via subsumption, such as medical device, procedure type, etc.

Cognitive grammar, a topic within cognitive linguistics, relates traditional grammar roles to a cognitive process [4]. As defined by the three general cognitive linguistics hypotheses, cognitive grammar involves the conceptualization from words in a grammatical unit (i.e., a sentence).

In general the conceptualization objectives of cognitive grammar is aimed at understanding two things [4]:

1. Things – cognitive notions that are usually nouns
2. Relations – cognitive notions that are usually verbs and adjectives

These two goals of cognitive grammar are functionally equivalent to the two core concept types in ontologies: 1) identify conceptual entities, and 2) identify relations between concepts (see Figure 1). Loosely speaking, item 1 is related to concept-based search, and item 2 is related to context-based search.

For example, if a sentence states that someone buys something, the cognitive grammar typically refers to participants and a relation, where the relation type matches the word used to describe the relation ('buy' in this case). Identifying the specific relation concept that maps to this role for a particular knowledge domain, however, requires more analysis.

For example, a spinal fusion may be accomplished without use of the Harrington rod. This simple negation operator is quite obvious to most anyone, i.e., this rod type is not applicable. But lacking the specific participant-role-participant relationship makes it difficult to determine this.

Unfortunately the field of cognitive grammar does not yet possess the computational approaches to the extent found in computational linguistics. For example, computational linguistics tools exist for mapping the linguistic part-of-speech to each words and phrase in a sentence. Although research has proceeded in cognitive linguistics in similar areas, algorithms and tools do not yet exist for the automated application of cognitive grammar. Hence it is not possible yet to parse sentences using a completely cognitive approach, and from this extract entities and relationships that map to ontological concepts (i.e., identify conceptual relations between concepts).

It is possible to map a noun phrase to one or more concepts using the MetaMap tool. Hence the remainder of this paper focuses on cognitive search limited to nouns and noun phrases. The development of computational approaches that automate cognitive grammar analysis and identify the cognitive relation between concepts is the topic of future research.

III. TOPOLOGY COVERING SPACE AND COGNITIVE RELEVANCE

The overarching objective of this research was to identify a *simple* measure of cognitive relevancy for ranking search results that improved precision. In addition, wherever possible minimizing the use of heuristics was desired to aid in diagnosing and fixing shortcomings.

The first objective of this section is to define a topology covering space for the ontology. The ontology will consist of a set of concepts and relationships that can be represented as a directed, acyclic graph whose highest level of abstraction consists of a small set of concepts. Relationship types are not restricted other than the set must include subsumption, as previously discussed. Furthermore, each concept name in the ontology must be unique. While name uniqueness is not a theoretical requirement for defining the covering space, it was required for practical purposes.

The second objective is to define a measure for comparing two covering spaces. This measure must be as simple as possible and reflect the cognitive relationships between two conceptual covering spaces defined by: a) the concepts associated with the search criteria, and, b) the concepts associated with the text being searched.

A. *Topology Space and Ontology Neighborhood*

Addressing the use of topology theory and neighborhoods applied to ontologies defines the mathematical basis for the cognitive search relevancy measure. It also addresses the intuitive relationship that relevancy has to cognition theory. These two areas provide the substantiation of an approach that minimizes heuristics and provides a simple measure for relevancy that achieves the basic objectives for this research.

The term neighborhood used in this paper refers to an ontological neighborhood of concepts where concepts in the same neighborhood share a set of cognitive notions of interest.

Definition 0: Per Willard [20], a topology on a set X is a collection τ of subsets of X, called the open set, satisfying the following:

1. Any union of elements of τ belong to τ,
2. Any finite intersection of elements of τ belong to τ,
3. Ø and X belong to τ

Definition 1: An ontology can be represented as a directed acyclic graph that consists of a set of vertices, V, and a set of relationships, R, where each relationship is a directed connection between two concepts c_{from} and c_{to}.

$$Ontology := graph\ DAG(V, R)\ where \quad (4)$$

$$V = \{c_1, c_2, \dots c_n\} \quad (5)$$

$$R = \{c_{from}, c_{to} | c_{from} \in V, c_{to} \in V\} \quad (6)$$

Definition 2: the set X used to define a topology for an ontology consists of all concepts in the ontology, i.e., V.

Definition 3: the distance d between two concepts in the ontology is the length of the shortest path P between the two concepts in the directed graph regardless of relationship type.

$$P(c_{from}, c_{to})\ is\ ordered\ set\ \{c_{from}, c_2, \dots c_{to}\} \quad (7)$$

$$d(c_{from}, c_{to}) = argmin_{P(c_{from}, c_{to})} (|P(c_{from}, c_{to})|) \quad (8)$$

Definition 4 the ontology neighborhood N for a concept c_{from} consists of itself plus any concept c_n where $d(c_{from}, c_n) > 0$.

$$N_{ont}(c_{from}) = \{c_{from}, c_n | d(c_{from}, c_n) > 0\} \quad (9)$$

Definition 3 is intended to reflect the basic notion of subsumption, i.e., specificity increases as the distance increases to higher level, more abstract concepts. Non-subsumptive relationships are included, per Definition 3 and in accordance with the ontological property that all concepts exist in at least one subsumptive hierarchy.

Use of non-subsumptive relationships can be justified by example using the relationship between spinal fusion and Harrington rod. This is a ‘uses’ relationship, not an ‘is a’ relationship. The use of the Harrington rod cognitively triggers the neuronal code for this medical device which in turn triggers the downstream feed-forward knowledge network in the cerebral cortex related to it. It therefore includes all of the concepts related to the Harrington rod, including the higher-level, abstract cognitive notions related to a device of this type. This is intuitively obvious in this case since the name of the concept includes the term Harrington rod, but such naming is not mandatory in the ontology.

Sophisticated search criteria may exclude certain concepts normally part of a concept’s neighborhood. But, as previously stated, use of predicate logic extracted from cognitive relations in sentences is for future research, so this functionality is excluded from the scope of this paper.

Definition 5: the size of the neighborhood for a concept c_{from} is the cardinality of its neighborhood set $N_{ont}(c_{from})$.

Definition 6: the neighborhood for multiple concepts is the union of the neighborhood set N associated with each concept.

$$N_{ont}(c_1, c_2, \dots c_m) = \bigcup_{i=1}^{m} N_{ont}(c_i) \quad (10)$$

Definition 7: the neighborhood common to two or more neighborhoods consists of the set of concepts found in the intersection of their neighborhoods.

$$N_I(N_1, N_2, \dots N_m) = \bigcap_{i=1}^{m} N_i \quad (11)$$

Definitions 1 through 7 define neighborhoods that contain subsets of the ontology concepts set V. The set V corresponds to X in Definition 0, whereas the set X contains the concepts for the entire knowledge domain defined in the ontology. The collection of neighborhoods N and intersections N_I corresponds to the collection of subsets τ referenced in Definition 0. This includes the empty set Ø and the entire domain ontology X. Therefore the use of Definitions 1 – 7 creates a topology over the ontology per the requirements of Definition 0.

This topology appears consistent with the cognitive process described in the prior section *B. Theory of Cognition*. Suppose a neural code is activated that represents concept c_{from}. And also suppose that this occurs in a hypothetical person whose knowledge base is complete and accurate. The activated neural code triggers the feedforward paths in the knowledge base of the cerebral cortex. This in turn activates a neighborhood of neural codes in the cerebral cortex, $N_{cortex}(c_{from})$, that are cognitively related. For our purposes, $N_{cortex}(c_{from})$ is interpreted to be a neighborhood of concepts represented by these activated neural codes. This, of course, is a simplification of the actual cognitive process.

This does not imply that the ontology $N_{ont}(c_{from})$ neighborhood is a 1:1 match with the $N_{cortex}(c_{from})$ neighborhood. The cerebral cortex $N_{cortex}(c_{from})$ and ontology $N_{ont}(c_{from})$ neighborhoods are viewed as functionally equivalent. Such equivalency is considered in respect to validating that the ontology topology T_{ont}, defined for determining cognition-base relevancy, is consistent with what is known about human cognition.

Definition 7: the neighborhood of the search criteria, $N_{criteria}$,consists of the union of the neighborhoods for each concept in the criteria. Likewise, the neighborhood of the text being searched, N_{text}, consists of the union of the neighborhoods for each concept found in the text.

Definition 8: a shared cognitive space for two or more neighborhoods consists of the intersection of these neighborhoods.

Definition 9: topology-based relevance,r, is measured by the relative size of cognitive space that the text neighborhood shares with the criteria neighborhood.

$$r = {|N_{criteria} \cap N_{text}|}\Big/{|N_{criteria}|} \qquad (12)$$

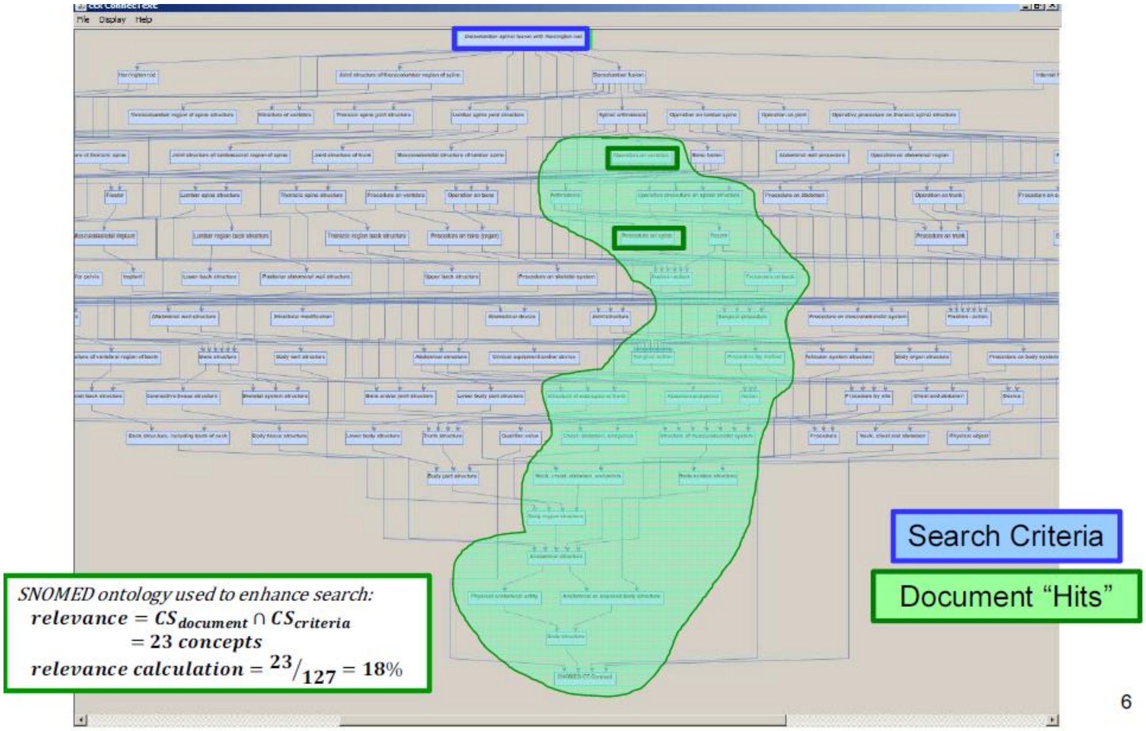

SNOMED ontology used to enhance search:
$relevance = CS_{document} \cap CS_{criteria}$
$= 23\ concepts$
$relevance\ calculation = {}^{23}/_{127} = 18\%$
Search Criteria
Document "Hits"
6

Figure 6: EXAMPLE OF SEARCH AND DOCUMENT NEIGHBORHOODS – suppose the search criteria consist of one concept, dorsolumbar spinal fusion with Harrington rod (blue). Suppose a document includes two related concepts. The green area is the covering space for the document. Cognitive relevance is the 23 concepts in the document neighborhood divided by the 127 concepts in the criteria neighborhood.

Cognitive relevance has range $0 \leq r \leq 1$ that indicates the relative size of the shared cognitive space that the text has with the criteria.

B. Example: Cognition versus Keyword

Consider two keywords – hypoglycemia and diabetes. In the context of keyword search these two terms have no relationship.

When considered within a cognitive context, however, they share a common parent, disorder of glucose metabolism.

Consider a search criteria containing diabetes. Some documents in the corpora being searched may contain hypoglycemia but not diabetes. These documents, while not having a cognitive relevancy equal to 100%, should, nonetheless, have a relevancy higher than concepts that exist in a completely different part of the ontology hierarchy. When criteria contain many concepts, such "near-miss" scenarios will likely return a set of documents of far greater relevance than keyword approaches.

C. Cogency-Based Relevance Measure

The cogency-based measure is a straight-forward application of confabulation theory [1-3]. The conditional probability of two concepts co-occurring in the same document and in the search criteria is the basis for computing cogency, as follows:

The search criteria consist of concepts extracted from the description supplied in natural language (for our experiments the search criteria was the patient's profile, along with search phrases provided by the physician.

$$Criteria = \{c_1, c_2, c_3 \ldots c_n | c_n \in C_{ontol}\} \quad (13)$$

The corpus consists of documents to be searched. In our experiments the corpus is a set of abstracts retrieved using the NLM's PubMed search tool and the search phrases from the physician.

$$Corpus = \{d_1, d_2, d_3 \ldots d_n\} \quad (14)$$

Each document in the corpus, i.e., abstract in our experiments, is represented by a set of concepts. These were identified using the MetaMap automated concept-text mapper.

$$d_{n,concepts} = \{c_1, c_2, c_3 \ldots c_m | c_m \in C_{ontol}, d_n \in Corpus\} \quad (15)$$

The corpus of concepts consists of the union of all concepts across all documents in the corpus.

$$Corpus_{concepts} = \{c_1, c_2, c_3 \ldots c_l | c_l \in C_{ontol}, \forall d_n \in Corpus\} \quad (16)$$

Cogency values are calculated using the frequency of occurrence of concept pairs across the entire corpus.

$$cogency_{c_l, c_m} = ln\big(prob(c_l | c_m)\big| c_l, c_m \in Corpus_{concepts}\big) \quad (17)$$

The cogency for a document is the sum of the cogency for all concept pairs found in both the document and the search criteria.

$$cogency_{d_n} = \sum ln\big(cogency_{c_l, c_m} \big| c_l, c_m \in Criteria\big) \quad (18)$$

This forms a distance measure for ranking search results, as a relation between documents where documents with a larger cogency value are ranked higher.

$$relation\ R: cogency_{d_a} > cogency_{d_b} \Rightarrow d_a R d_b (19)$$

III. TEST AND RESULTS

A. *Approach*

1) *Purpose and Summary*

The purpose testing for this research is exploratory in nature. The question is whether or not the two cognitive relevance measures is a plausible approach for improved search precision, and hence, suitable for continued research and validation.

A comparison was made between the cognitive search approach described herein and the traditional keyword approach. The baseline for comparison is the NLM's PubMed search tool, a popular search tool in medicine. This provides the following:

- Provide a baseline for keyword search precision for comparison.
- Extract a large corpus of document abstracts for search using the cognitive relevance measure.

Precision results for the cognitive approaches are compared to the results provided by the PubMed traditional keyword approach. The recall measure is not used since a corpus that identifies all relevant documents was not practical due to storage limitations. However, the corpus size for computing cognitive search precision was large enough to be suitable.

Real-world patient profiles were used for testing. These consisted of the History of Present Illness, or HPI, a clinical artifact created in a number of clinical processes. The HPIs used for this study are fictitious, but, reflect the clinical experience of a cardio-thoracic surgeon and hence realistic.

A total of ten HPIs were used, which, while appearing to be a relatively small sample size, was deemed appropriate for exploratory testing since 20 NLM abstracts are scored for each patient and precision is computed for the aggregate of all abstracts across all HPIs. This provided a total of 200 abstracts for dichotomous categorization by the physician (either relevant or not relevant) and precision calculations. The HPIs ranged from 1 to 5 paragraphs in length. See Figure 7 for an example. Sample size justification and other statistical considerations are provided in the Results section.

The scenario used for testing is that of a surgeon who needs to identify any clinical, procedure, device, or other medical factors that may increase safety or clinical outcomes risk for that particular patient. The search tool supports this task with a broad-based search to perform a "sweep" of potential factors that the surgeon must address to ensure minimal risk and optimal results. It is envisioned that this would occur as part of the normal clinical process that evaluates and plans the surgical procedure. That is, this scenario is envisioned as standard step in the typical clinical process. The purpose is to include the effectiveness of the unique capabilities enabled by the cognitive relevance measure. That is, a natural language interface for a fast and easy search interface, analysis of all relevant patient information via use of the entire HPI as the search criteria, and improved confidence in results due to higher search precision.

A 64-year-old women presents with a 3 cm mass in her left upper lobe, which was not present 18 months previously. Computed tomography confirms the presence of the mass without evidence of mediastinal adenopathy. Transthoracic fine needle aspiration reveals non-small cell lung cancer. The surgeon reviews the patient’s medical record, x-ray findings, pulmonary function studies, laboratory results, and bronchoscopy report. A mediastinoscopy has been performed which shows no evidence of N2 or N3 nodal involvement. Informed consent is obtained. The planned procedure is discussed with the anesthesiologist.

The patient is admitted to the hospital the morning of the scheduled operation and undergoes a left posterolateral thoracotomy. The lateral chest wall, diaphragm, pericardium, and mediastinum are examined for evidence of metastatic disease; if detected, appropriate biopsies are obtained. The pulmonary ligament is divided and representative pulmonary ligament, paraesophageal, aortopulmonary window, subcarinal, and hilar lymph nodes are sampled. The pulmonary artery is exposed in the fissure and the fissure is completed with a stapler. The segmental pulmonary arteries to the upper lobe are isolated, ligated, and divided. The superior pulmonary vein is isolated, divided and over-sewn. The distal vein is ligated. All peribronchial tissue and lymph nodes are reflected into the specimen and the left upper lobe bronchus is isolated and divided at its origin. The closed bronchial stump is checked for competency. Two chest tubes are inserted into the pleural cavity and the thoracotomy is closed. The patient is extubated and sent to the post anesthesia recovery unit.

Chest tubes are removed in the hospital on the third or fourth day. Following discharge on the seventh postoperative day, the patient is seen in the office for suture removal and checking of the incision site, chest x-ray, and management of routine postoperative problems with pain management, wound care, and return of preoperative pulmonary and physical function

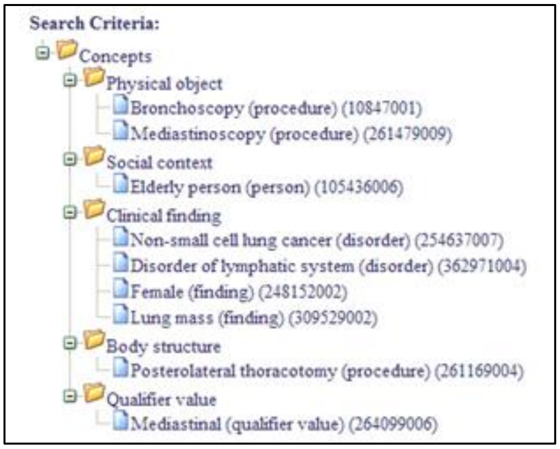
Figure 7: HISTORY OF PRESENT ILLNESS – example of the HPI for a patient undergoing cancer surgery. Surgical concerns typically stem from complications that can occur due to patient condition and medical history. Performing a fast and accurate

search for risk factors assists the surgeon in taking steps to mitigate risks and improve clinical outcomes.

The study approach is blind. The physician does not know the search approach used to retrieve the abstract he categorizes, nor does he know the ranked relevance position of the abstract.

2) *Steps*

The search criteria, i.e., the HPI, were analyzed using the MetaMap program from the NLM. This parses the HPI into sentences, and then sentences into phrases. It then retrieves a list of concepts that match the words/phrases in the HPI. This list will be used to identify the criteria's cognitive neighborhood $N_{criteria}$. See Figure 8 for an example.



Search Criteria:

- Concepts
  - Physical object
    - Bronchoscopy (procedure) (10847001)
    - Mediastinoscopy (procedure) (261479009)
  - Social context
    - Elderly person (person) (105436006)
  - Clinical finding
    - Non-small cell lung cancer (disorder) (254637007)
    - Disorder of lymphatic system (disorder) (362971004)
    - Female (finding) (248152002)
    - Lung mass (finding) (309529002)
  - Body structure
    - Posterolateral thoracotomy (procedure) (261169004)
  - Qualifier value
    - Mediastinal (qualifier value) (264099006)

Figure 8: SEARCH CRITERIA – example of search criteria, i.e., list of concepts mapped to the History of Present Illness.

The physician also provided a set of keywords for each HPI. This was used to perform a search for each HPI using the PubMed search engine (as mentioned earlier, for computing a precision baseline and to develop search corpus).

A corpus was developed for the cognitive approach using the keywords supplied by the physician. All abstracts returned by the keyword approach were downloaded and stored in a relational database. There were 93,436 abstracts downloaded and included in the corpus for our experiments.

Concept recognition for each abstract in the corpus was then performed using MetaMap. Concept maps provided by MetaMap were stored in the relational database, i.e., each NLM abstract is associated with a cognitive covering space based upon the concepts found in the abstract.

A desktop application was provided to the physician for scoring the top 20 NLM abstracts for each HPI. The 20 abstracts consisted of the top 10 ranked abstracts from keyword search, and the top 10 ranked abstracts from cognitive search. The categorization for the abstracts is blind, that is, the physician does not know which search method was used, and the order in which the abstracts are presented is randomized.

3) *Precision Calculations and Confidence Goal*

Precision was calculated as follows [21]:

$$precison = {}^{tp}/_{tp + fp} \tag{20}$$

$$\begin{aligned} &where: \\ &tp = true\ positives \\ &fp = false\ positives \end{aligned}$$

A target confidence level of 5% was chosen for Type I error for this study ($p = 0.05$).

Precision was based upon the first 10 abstracts returned by the search methods, rank ordered by relevancy. Basing precision upon the first 10 abstracts for each search method both provides the desired confidence level in study results, and also mimics the typical hectic clinical environment where accurate information is needed in a relatively short period of time. This approach is consistent with other findings [22].

4) *Null Hypothesis, Sample Size, and Normal Approximation to Binomial*

The null hypothesis is that the difference in the proportion of relevant documents is zero, as follows (all statistical equations from Devore [23]):

$$H_o: p_{cognitive} - p_{keyword} = \Delta_p = 0 \tag{21}$$

$$\begin{aligned} &where \\ &p_{cognitive} \\ &= precision\ of\ cognitive\ search \\ &p_{keyword} \\ &= precision\ of\ legacy\ keyword\ search \end{aligned}$$

The alternative hypothesis, H_a, is that the precision of concept-based search, in combination with use of the cognitive relevance distance measure, results in improved precision, as follows:

$$H_a: \Delta_p > 0 \tag{22}$$

The binomial probability distribution applies for the dichotomous experiments performed in this research, as follows:

$$b(x;n,p) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} \; where \; x = 0,1,2,\dots n \\ 0 \; otherwise \end{cases} \tag{23}$$

$where$

$x \; is \; number \; of \; successful \; trials$

$n \; is \; sample \; size$

$p \; is \; success \; proportion \; of \; population$

Of course the p for the binomial is not the same p value for confidence in test results.

The normal distribution will be used to approximate the binomial to simplify computation of the Type I error. The mean and standard deviation of the normal approximation, stated in terms of success proportion instead of number of successes, is as follows:

Mean of the binomial: $\mu_X = np$ (24)

Standard deviation of the binomial: $\sigma_X = \sqrt{np(1-p)}$

Since the measure of interest is the population success proportion p, Equations 12 and 13 are restated in terms of the success proportion p, as follows:

Mean of the binomial: $\mu_X = p$ (25)

Standard deviation of the binomial:

$$\sigma_X = \sqrt{p(1-p)/n} \tag{26}$$

The standard normal variable z is then stated in terms of ρ and n as follows:

Standard normal variable: $z = \frac{X-\mu_P}{\sigma_P} = \frac{X-p}{\sqrt{p(1-p)/n}}$(27)

The normal distribution is a suitable approximation to the binomial when two conditions are met in Devore, page 166 [23], as follows:

Rule 1. $np \geq 10$
Rule 2. $n(1-p) \geq 10$

That is, the expected number of successes should be at least 10, and the expected number of failures should be at least 10.

Solve n for both rules, as follows:

Rule 1. $n \geq {}^{10}/_{p}$
Rule 2. $n \geq {}^{10}/_{(1-p)}$

If the sample size, success proportion, and failure proportion for both methods of search complies with the two rules for using the normal approximation to the binomial, then the difference in these proportions can be approximated with the normal distribution (all equations below for difference in proportions from Devore pp. 391-397 [23]).

Rule 1. $n \geq \frac{10}{p_{cognitive}} \; and \; n \geq \frac{10}{p_{keyword}}$
Rule 2. $n \geq \frac{10}{(1-p_{cognitive})} \; and \; n \geq \frac{10}{(1-p_{keyword})}$

$where$

$n = sample \; size, each \; search \; method$

B. Results

1) Precision and p Value

The precision results shown in Table 2 are based upon the total true positives tp and total false positives fp across all four patient histories.(22)

Table 2: PRECISION RESULTS

Search Type	tp	fp	precision
Keyword	58	42	0.58
Topology-Based	40	60	0.40
Cogency-Based	1	99	0.01

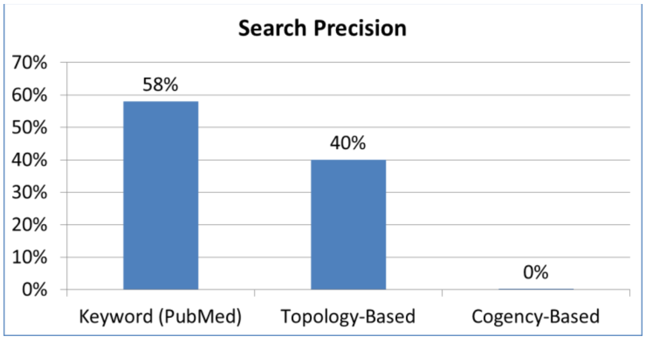


Figure 9: PRECISION RESULTS – PubMed outperformed the topology-based ranking measure, and the cogency-based measure was inadequate. This suggests that use of a cognitive-based ranking measure alone is insufficient to achieve significant improvements in search precision.

The p value for rejecting the null hypothesis of equal proportions is estimated using the normal distribution:

$$z = \frac{\hat{p}_{cognitive} - \hat{p}_{keyword} - (p_{cognitive} - p_{keyword})}{\sqrt{\frac{p_{cognitive}(1-p_{cognitive}) + p_{keyword}(1-p_{keyword})}{n}}} \tag{28}$$

$where$:
$n = sample\ size, each\ search\ method$

The tests resulted in $\hat{p}_{cognitive} = 0.404$ and $\hat{p}_{keyword} = 0.58$, or $\Delta_p = 0.18$. The standard normal distribution variable for this difference $z = 2.54$, or 0.5% significance, albeit at a rate lower than PubMed

IV. CONCLUSIONS

This paper has focused on the definition of the cognitive relevance measures and theoretical considerations in terms of consistency with concept and context-based search, consistency with a theory of cognition, and consistency with theories of cognitive grammar. Exploratory test results were also provided that suggest that the cognitive relevance measures, when used in isolation, are inadequate for improving search precision.

While use of concept-based approaches has potential for improved search precision, a simple ranking measure alone cannot address the complex cognitive functions required for high-precision search. . More complicated methods, such as neural networks, are likely required.

Despite the limited sample size, the difference in precision rates is statistically significant . . The need for improvements in cognitive-based methods is apparent from this.

These results are useful in the decision-making process for future research efforts. Our test results suggest that a greater emphasis on sophisticated approaches, for example neural networks, may pay a higher dividend in precision.

In addition, follow-on research into cognitive grammar appears promising with the goal of extracting query conditions from text that includes cognitive relationships (e.g., negation conditions, cross-phrase relationships for concept recognition, etc.). This addresses the fidelity/precision question: as more relationships are identified in the sentences, then improved fidelity in concept recognition likely occurs.

Domain ontologies, required when using the cognitive search measure, are typically built by hand and hence are laborious and expensive. This is a limiting factor for use of a cognitive relevance measure when the economic benefits of high-precision search are difficult to quantify, i.e., “soft”, in comparison to alternative uses of an organization’s capital. Given this context it appears reasonable to assume that until an automated approach is found that effects a material reduction in the cost of creating ontologies the use of the cognitive relevance measure will be limited to those domains where ontologies already exist.

References

[1] R. Hecht-Nielsen, "Cogent confabulation," Neural Networks, vol. 18, pp. 111-115, 3// 2005.

[2] R. Hecht-Nielsen, Confabulation Theory The Mechanism of Thought. LaJolla, California: Springer, 2007.

[3] S. Solari, A. Smith, R. Minnett, and R. Hecht-Nielsen, "Confabulation Theory," Physics of Life Reviews, vol. 5, pp. 106-120, 6// 2008.

[4] G. Radden and R. Dirven, Cognitive English Grammar. Amsterdam, The Netherlands: John Benjamins Publishing Company, 2007.

[5] N. L. o. Medicine. (2013). Unified Medical Language System (UMLS). Available: http://www.nlm.nih.gov/research/umls/quickstart.html

[6] A. Kumar, Y. L. Yip, B. Smith, and P. Grenon, "Bridging the gap between medical and bioinformatics: An ontological case study in colon carcinoma," Computers in Biology and Medicine, vol. 36, pp. 694-711, 7// 2006.

[7] D. Movshovitz-Attias, S. Euijong Whang, N. Noy, and A. Halevy, "Discovering Subsumption Relationships for Web-Based Ontologies," presented at the Proceedings of the 18th International Workshop on Web and Databases, Melbourne, VIC, Australia, 2010.

[8] M. Bastiaansen and P. Hagoort, "Oscillatory neuronal dynamics during language comprehension," in Progress in Brain Research. vol. Volume 159, K. Christa Neuper and Wolfgang, Ed., ed: Elsevier, 2006, pp. 179-196.

[9] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition. Cambridge, United Kingdom: Cambridge University Press, 2014.

[10] R. Hecht-Nielsen, "The Mechanism of Thought," in Neural Networks, 2006. IJCNN '06. International Joint Conference on, 2006, pp. 419-426.

[11] A. G. Huth, W. A. de Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant, "Natural speech reveals the semantic maps that tile human cerebral cortex," Nature, vol. 532, pp. 453-458, 04/28/print 2016.

[12] R. Moskovitch, S. B. Martins, E. Behiri, A. Weiss, and Y. Shahar, "A Comparative Evaluation of Full-text, Concept-based, and Context-sensitive Search," Journal of the American Medical Informatics Association : JAMIA, vol. 14, pp. 164-174, Mar-Apr

[13] A. R. Aronson, "The effect of textual variation on concept based information retrieval," Proceedings : a conference of the

American Medical Informatics Association / ... AMIA Annual Fall Symposium. AMIA Fall Symposium, pp. 373-377, // 1996.

[14] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," Proceedings / AMIA . Annual Symposium. AMIA Symposium, pp. 17-21, // 2001.

[15] A. R. Aronson, "MetaMap: Mapping Text to the UMLS Metathesaurus," UMLS White Paper, 2006.

[16] A. Aronson, "The Current State of MetaMap and MMTX," 2009.

[17] N. H. Shah, N. Bhatia, C. Jonquet, D. Rubin, A. P. Chiang, and M. A. Musen, "Comparison of concept recognizers for building the Open Biomedical Annotator," BMC Bioinformatics, vol. 10, pp. S14-S14, 09/17 2009.

[18] A. R. Aronson and F. M. Lang, "An overview of MetaMap: Historical perspective and recent advances," Journal of the American Medical Informatics Association, vol. 17, pp. 229-236, 2010.

[19] W. Croft and D. A. Cruse, Cognitive Linguistics: Cambridge University Press, 2004.

[20] S. Willard, General Topology. Mineola, New York: Dover Publications, Inc., 1970.

[21] C. D. Manning and H. Schütze Foundations of Statistical Natural Language Processing. Cambridge, Massachusetts: The MIT Press, 1999.

[22] Z. S. Shariff, A. D. S. Bejaimal, M. J. Sontrop, V. A. Iansavichus, B. R. Haynes, A. M. Weir, et al., "Retrieving Clinical Evidence: A Comparison of PubMed and Google Scholar for Quick Clinical Searches," J Med Internet Res, vol. 15, p. e164, 08/15 2013.

[23] J. Devore, Probability and Statistics for Engineering and the Sciences. Boston, MA: Cengage Learning, 2016.

APPENDIX D

COGNITIVE RELEVANCE TEST PLAN

Test Plan

Cognitive Relevance Measure

George Shannon
PhD Candidate, Systems Engineering
Missouri University of Science and Technology

Version 2.0

February 26, 2017

1. Content of Plan

This plan contains objectives, definitions, background, measures, any preliminary results relevant to testing, calculations and relevant statistical approaches or theories, and references used. Emphasis is placed on documentation of important details for review by others.

2. Objective

Quantify improvement in search precision using a cognitive-based search approach. Determine if precision has improved, and by how much at a specified level of statistical significance.

This includes use of the following technologies:

1. Use of MetaMap for automated concept recognition in text.
2. Use of the cognitive relevance distance measure for rank ordering search results by relevance.

MetaMap is used for concept recognition for consistency with prior test results and to isolate the effect of using the inverse ontology cogency approach. The latter goal is aimed at future testing.

This will be a blind study. The physician performing the testing will not know which search method was used to retrieve a document he is categorizing, and the order of documents presented to the physician will be randomized.

3. Definition

Cognitive-based search is defined to include the following three features: 1) the recognition of concepts in text, 2) retrieving documents containing the desired concepts, and, 3) rank-ordering these documents using the concept relevance distance measure.

NLM – National Library of Medicine, including PubMed, the keyword search engine used by NLM.

4. References

Statistical theories, methods, equations, etc. come from Devore, 2016 [1].

5. Background

5.1 Search Product Niche

The purpose of the cognition-based approach is to provide high-precision search at a level that materially differentiates the concept-base search from legacy keyword search.

It is *not* intended to replace a general, Internet-based search engine, such as Google or Bing.

5.2 Preliminary Results

Preliminary tests, using four (4) – history of present illness (HPI) documents, resulted in the following:

Precision for keyword search: $p_{keyword} = 0.20\ \pm$

Precision for cognitive search: $p_{cognitive} = 0.70\ \pm$

These results are preliminary.

6. Statistical Tests

Quantify the improvement in precision when using a concept-based search approach in comparison to legacy keyword-based approaches.

6.1 Search Precision Measurement

Precision, ρ, is defined as follows:

$$p = {}^{tp}/_{tp + fp} \tag{1}$$

$where$

$tp = true\ positives, and$

$fp = false\ positives.$

6.2 Null and Alternative Hypotheses

Null hypothesis H_o: the precision of concept-based search, in combination with use of the cognitive relevance measure, is no different than the precision of legacy approaches.

$$H_o: p_{cognitive} - p_{keyword} = \Delta_p = 0 \tag{2}$$

$where$

$p_{cognitive} = precision\ of\ cognitive - based\ search$

$p_{keyword} = precision\ of\ legacy\ keyword\ search.$

Alternative hypothesis H_a: the precision of concept-based search, in combination with use of the cognitive relevance distance measure, results in improved precision.

$$H_a: \Delta_p > 0 \tag{3}$$

6.3 Computations

6.3.1Type I and Type II Test Significance Objectives

The targeted level of significance for the Type I error, α, the probability of rejecting H_o when it is true, is as follows:

$$\alpha = 0.05 \tag{4}$$

The targeted level of significance for the Type II error, β, the probability of failing to reject H_o when it is false, is as follows:

$$\beta = 0.10 \tag{5}$$

The actual Type I and Type II probabilities will be computed after testing is complete and compared to these objectives.

6.3.2 Binomial Distribution for Test Samples

Each document returned by the two search methods (legacy keyword and cognitive-based) for each HPI will be binary scored as relevant or not by a physician.

An experiment with the following characteristics will follow the binomial discrete distribution (Devore pg. 119):

a. Experiment has two possible outcomes (success or failure),
b. The probability of each of these outcomes is constant for all samples, and,
c. Each sample is independent.

The tests being performed meet these criteria.

The binomial distribution is defined as follows:

$$b(x;n,p) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} \ \ where\ x = 0,1,2, \dots n \\ \qquad\qquad 0\ otherwise \end{cases} \tag{6}$$

$x\ is\ number\ of\ successful\ trials$

$n\ is\ sample\ size$

$p\ is\ success\ proportion\ of\ population$

6.3.3 Normal Approximation and Sample Size Range

The normal distribution will be used to approximate the binomial to simplify computation of the Type I error. The mean and standard deviation of the normal approximation, stated in terms of success proportion instead of number of successes, is as follows:

Mean of the binomial: $\mu_X = np$ (7)

Standard deviation of the binomial: $\sigma_X = \sqrt{np(1-p)}$ (8)

Since the measure of interest is the population success proportion p, Equations 7 and 8 are restated in terms of the success proportion p, as follows:

Mean of the binomial: $\mu_X = p$ (9)

Standard deviation of the binomial: $\sigma_X = \sqrt{p(1-p)/n}$ (10)

The standard normal variable z is then stated in terms of ρ and n as follows:

Standard normal variable: $z = \frac{X-\mu_P}{\sigma_P} = \frac{X-p}{\sqrt{p(1-p)/n}}$ (11)

The normal distribution is a suitable approximation to the binomial when two conditions are met (Devore pg. 166), as follows:

Rule 3. $np \geq 10$
Rule 4. $n(1-p) \geq 10$

That is, the expected number of successes should be at least 10, and the expected number of failures should be at least 10.

Solve n for both rules, as follows:

Rule 1. $n \geq {}^{10}/_{p}$
Rule 2. $n \geq {}^{10}/_{(1-p)}$

The null hypothesis for the difference of two population proportions is shown in Equation 2, that is, there is no difference. If the sample size, success proportion, and failure proportion for both methods of search complies with the two rules for using the normal approximation to the binomial, then the difference in these proportions can be approximated with the normal distribution (all equations below for difference in proportions are from Devore pp. 391-397).

Rule 1. $n \geq {}^{10}/_{p_{cognitive}}\ and\ n \geq {}^{10}/_{p_{keyword}}$
Rule 2. $n \geq {}^{10}/_{(1-p_{cognitive})}\ and\ n \geq {}^{10}/_{(1-p_{keyword})}$

$where$

$n = sample\ size, for\ cognitive\ search\ and\ keyword\ search, each$

6.3.4 Difference in Proportions

The population mean and variance are estimated using the sample data, as follows:

$$\hat{p}_{cognitive} = {}^{X}\!/_{n} \tag{12}$$

where X is number of success for cognitive search, and $X \sim Bin(n, p_{cognitive})$

$$\hat{p}_{keyword} = {}^{Y}\!/_{n} \tag{13}$$

where Y is number of successes for keywork search, and $Y \sim Bin(n, p_{keyword})$

$$E(\hat{p}_{cognitive} - \hat{p}_{keyword}) = p_{cognitive} - p_{keyword} \tag{14}$$

$$V(\hat{p}_{cognitive} - \hat{p}_{keyword}) = \frac{p_{cognitive}(1-p_{cognitive}) + p_{keyword}(1-p_{keyword})}{n} \tag{15}$$

If the inequalities specified in Rules 1 and 2 are met, then both $p_{cognitive}$ and $p_{keyword}$ can be approximated by the normal distribution. In this case the standardized z value is as follows:

$$z = \frac{\hat{p}_{cognitive} - \hat{p}_{keyword} - (p_{cognitive} - p_{keyword})}{\sqrt{\frac{p_{cognitive}(1-p_{cognitive}) + p_{keyword}(1-p_{keyword})}{n}}} \tag{16}$$

If the null hypothesis holds, then $\Delta_p = 0$. The standardized z value is then as follows:

$$z = \frac{\hat{p}_{cognitive} - \hat{p}_{keyword} - 0}{\sqrt{\frac{2\hat{p}(1-\hat{p})}{n}}} \tag{17}$$

where the combined sample estimate of the population proportion $\hat{p}$ is

$$\hat{p} = \frac{X+Y}{2n} \tag{18}$$

The test significance, α, will be the area under the normal curve to the right of the z value computed by Equation 17, i.e., this is an upper-tail test.

6.3.5 Confidence Interval under Alternative Hypothesis

If the null hypothesis is shown to be false, at the desired significance level $\alpha = 0.05$, then a confidence interval for the difference in proportions will be computed.

Under the alternative hypothesis H_a, the variance of Δ_p is no longer pooled since equality of proportions, the null hypothesis of $\Delta_p = 0$, has been rejected. Instead, the variance is computed per Equation 18. The confidence limits for Δ_p becomes a two-tail test as follows:

$$\widehat{\Delta}_p \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_{cognitive}(1-\hat{p}_{cognitive}) + \hat{p}_{keyword}(1-\hat{p}_{keyword})}{n}} \tag{19}$$

Using a P-value equal to the Type I error $\alpha = 0.05$, $z_{\alpha/2} = 1.96$.

6.4 Sample Size

6.4.1 Minimum Sample Size for Normal Approximation

For a preliminary estimate of the minimum sample size for use of the normal distribution as an approximation to the binomial, the preliminary results are used. The expectation is that tests per this plan will be approximately the same.

Assuming that similar values obtained during preliminary testing are obtained, i.e., $p_{keyword} = 0.20$ and $p_{cognitive} = 0.70$, then a minimum sample size can be computed using Equations 13 and 14.

For cognitive search:

$$n \geq {}^{10}/_{0.70} \; and \; n \geq {}^{10}/_{(1-0.70)}, or \; n \geq 15 \; and \; n \geq 33 \tag{23}$$

For keyword search:

$$n \geq {}^{10}/_{0.20} \; and \; n \geq {}^{10}/_{(1-0.20)}, or \; n \geq 50 \; and \; n \geq 13 \tag{24}$$

Equation 24 indicates that a minimum sample size $n \geq 50$ is required to use the normal distribution to approximate the binomial.

6.4.2 Sample Size Considering Type I and Type II Error Objectives

Equation 9.7 of Devore [1] provides an estimate of the sample size when taking into account the Type I and Type II error probabilities, as follows:

$$n = \frac{\left[z_{\alpha}\sqrt{(p_{cognitive}+p_{keyword})(q_{cognitive}+q_{keyword})/2}+z_{\beta}\sqrt{p_{cognitive}q_{cognitive}+p_{keyword}q_{keyword}}\right]^{2}}{\Delta_{p}^{2}} \tag{25}$$

$$where$$

$$q_{cognitive} = 1 - p_{cognitive} \; and \; q_{keyword} = 1 - p_{keyword}$$

Note that the sample size n is equal for *both* populations. That is, a total of n documents are scored for the cognitive search approach, and a total of n documents are scored for the keyword search approach.

With a Type I and Type II error probabilities $\alpha = 0.05$ and $\beta = 0.1$ respectively, $z_{\alpha} = 1.645$ and $z_{\beta} = 1.28$. The estimated sample size, using the preliminary success rations $p_{cognition} = 0.70$ and $p_{keyword} = 0.20$, is as follows:

$$n = \frac{\left[1.645\sqrt{(0.5)(0.90)(1.10)} + 1.28\sqrt{(0.7)(0.3) + (0.8)(0.2)}\right]^{2}}{(0.70 - 0.20)^{2}}$$

$$n = [(1.1574 + 0.7786)/0.50]^{2} = 15$$

6.4.3 Patient Count and Retrieved Document Count

The sample size calculations in Sections 0 and 0 indicate that the minimum sample size is 50. This sample size is driven by the requirements that must be met to approximate the binomial distribution with the normal.

This sample size refers to the number of search documents scored by a physician as relevant to a patient's history, i.e., success is defined as relevance to a patient.

The next step is to determine the number of patient histories to use and number of documents scored per patient such that the total number of scored documents is greater than or equal to 50.

The approach taken select the number of patient histories and number of scored documents per patient is as follows:

1. Select the total number of scored documents n to obtain from a physician for each of the populations, and meets the constraint that $n \geq 50$.

2. Identify the minimum number of documents per patient that the physician scores. This number will replicate what likely occurs in practice. That is, it is the maximum number of documents that a busy clinician will review to: a) get the desired information, or, b) feel confident that the information is not available. In other words, determine the maximum number of documents that a physician will tolerate to obtain the desired information.
3. Calculate the number of patient histories necessary to obtain the desired total number of scored documents.

Results:

1. Number of documents for each population: $n = 100$.
2. Min. number of documents per patient: 10.
3. Number of patient histories: 10.

This approach obtains more patient histories than used in the preliminary testing. The small number of patient histories in the preliminary testing tended to raise questions about sample size.

Moreover, since the preliminary testing indicated a relatively large improvement in precision, $\Delta_p = 0.50$, a large sample size is less of an issue.

A summary of the sample size analysis is provided in Table 1.

Table 1: Sample Size Summary

Sample Size Selection	Value
Number of documents for each population	100
Number of documents scored per patient	10
Number of patient histories	10
Targeted Type I error	$\alpha \leq 0.50$
Targeted Type II error	$\beta \leq 0.10$

7. Test Steps and Logistics

The process used for preliminary testing will be repeated for this test plan, but with larger sample size and more sophisticated outcomes analysis.

This process is shown in Table 2 below.

Table 2: Steps for Testing

Who	What	How/Why/Deliverable
Physician	Identify clinical need and search scenario	• Determine what to search for. Select a typical clinical scenario where a high-precision search tool can help improve patient safety and other clinical outcomes. • Ideally the medical opinion used to determine relevance of each document scored is based upon this clinical scenario.

Who	What	How/Why/Deliverable
Physician	Select patient histories.	Identify history of present illness (HPI) for the number of patients shown in **Table 1**, Section 0 above.
Physician	De-identify HPIs for HIPAA compliance	Ensure all patient identifiers are removed from HPIs.
Physician	Identify keywords	• For each HPI, indicate keywords to use for the keyword-based search. • Can simply create a MS Word document or Excel spreadsheet that contains the HPIs, and for each HPI, the keywords to use.
Physician	Email HPIs and keywords to PhD student	Email file from prior step to author
PhD student	Import into desktop app	Import the HPIs into a desktop application that in later steps will be used by physician to record score results.[2]
PhD student	Perform automated concept recognition	Execute the MetaMap process to identify concepts in the HPIs.
PhD student	Perform keyword search	• Use PubMed and the keywords provided by the physician to execute keyword searches for each HPI. A very large set of documents are retrieved for each HPI using keyword search so that the cognitive-based search has a large corpora from which to select highly-relevant documents. • Note that PubMed will return the documents rank ordered. However, a small number of the top-ranked documents (per sample size in **Table 1**) are provided to the physician for relevance scoring using the keyword search approach. • Retrieving a large number of documents is simply to avoid having to download, index, and store the entire NLM database for search. • Import results into the database for the desktop application.

[2] The desktop application will make it easy for the physician to read and score search results for each HPI. A portable database will be used by this application such that the physician can transmit these scores back to the author by simply emailing the database file. The author will then use this database to determine precision results.

Who	**What**	**How/Why/Deliverable**
PhD student	Perform cognitive-based search	• Execute cognition-based search to retrieve candidate documents from those downloaded from PubMed, and then rank order results using the cognitive-based relevance measure. • Import results into the database for the desktop application, limited to the number of documents per patient as shown in **Table 1**.
PhD student	Send documents and desktop app to physician	• Package database and desktop application into a single file that can be installed by simply copying to a computer. • Provide instructions/readme text, along with any other suitable documentation if required.
Physician	Score documents	• Follow simple forms provided in the desktop application to review documents returned by the two desktop for each HPI. [3] • For each patient history, the desktop application will provide a list of documents to for the physician to categorize as "relevant" or "not relevant." • This list will be randomized and no indication is provided to the physician as to the search method used to extract each document, i.e., this is a blind study. • The form will provide the ability to view the HPI concurrently with viewing each document. • To "score" a document, physician simply clicks on a field that indicates whether or not this document is relevant.
Physician	Email desktop app database to PhD student	Copy the database file for the desktop application and email to the author. Detailed instructions will be provided with the application.
PhD student	Perform analysis	• Import scores and perform analysis in accordance with this test plan. • Report results to physician and academic advisers.. • Update journal article (currently in draft). • Obtain final reviews by Grad Office. • Obtain final reviews by physician and academic advisers. • Submit paper to journal.

[3] A "document" as referred to in this plan is an abstract in the NLM. Full-text documents are not used since full-text is not available for all abstracts.

References

1. Devore, J., Probability and Statistics for Engineering and the Sciences. 2016, Boston, MA: Cengage Learning.

APPENDIX E

BIOGRAPHY, JAMES LEVETT, MD

Dr. Levett is providing the medical opinion on relevancy of search results, as discussed in the test plan in Appendix D. He has extensive experience in medical research and new healthcare technologies.

Dr. James Levett is the Chief Medical Officer of Physicians' Clinic of Iowa. Dr. Levett has maintained and active practice in adult cardiac, vascular, and thoracic surgery for the past 25 years, and is actively working in the areas of process management excellence, outcomes research, and the implementation of quality management system principles in healthcare organizations. In 2003, Dr. Levett led PCI to become certified to ISO 9001:2000, the largest medical group in the U.S. to achieve this distinction.

Dr. Levett serves on the Nomenclature and Coding Committee of the Society of Thoracic Surgeons and has recently been appointed the STS Advisor to the Relative Value Update Committee of the American Medical Association. He is a past president of the Iowa Society of Thoracic Surgeons, and served as a National Examiner for the Baldrige National Quality Award Program in 2003 and 2004. Dr. Levett is a member of the Iowa Healthcare Collaborative and the Wellmark Physicians Quality Council. He is also the principle investigator on a recently approved AHRQ grant, Partnerships in Implementing Patient Safety (PIPS), RFA HS-05-012; Project Title, "Improving Warfarin Management in Competitive Healthcare Using ISO 9001 Principles." The two-year grant will establish an anticoagulation clinic in Cedar Rapids using ISO 9001 principles, and will test the concept of using ISO 9001 principles to improve healthcare within a community by allowing competing providers to work together using a common ISO framework.

Dr. Levett graduated cum laude from Carleton College, earned his medical degree from the University Of Iowa College Of Medicine, and completed surgical residencies in both general and thoracic surgery at the University of Chicago Hospitals and Clinics. He did post-graduate work in electrophysiology at Duke University Medical Center. Prior to returning to Iowa, Dr. Levett was Chairman of the Department of Surgery at Lutheran General Hospital in Park Ridge, IL. He is an author and/or contributing author of over 80 original articles, books, and scientific abstracts.

APPENDIX F

SEMANTIC PROCESSING SYSTEM

Semantic Processing System Framework

SPS

George J. Shannon

Abstract

Semantic processing of textual information involves the use of technologies across different disciplines – linguistics and natural language processing, computational intelligence, and high-speed information processing. Some but not all of the technologies required to provide fast and effective semantic processing in an economical fashion are found in these different disciplines. This paper will propose a distributed system framework for semantic processing, entitled Semantic Processing Framework or SPS. The intent of the SPS Framework is to provide a common language for identifying, developing, and evaluating required core technologies. In addition it will present a new element not yet apparent in other semantic applications that addresses the need for computing semantic relevance. It will also identify unfulfilled needs that are required for a semantic processing system to be economically viable and achieve performance needs. This framework will be presented using a case study from the development of a prototype semantic processing application ("medText") for information retrieval from medical research text and other examples. Based upon the knowledge gained from developing this application, gaps will be identified in currently available technologies that reduce the value of semantic processing systems. New products will be suggested that appear practical (i.e., economically viable and high performance) if these gaps are addressed.

I. Introduction and Background

Introduction

Semantic processing uses automated computer technologies to apply human reasoning in the interpretation of unstructured textual information. In essence it strives to create software applications that can mimic the complex and ambiguous/fuzzy logic that humans are capable of when reading text. Beginning at a young age humans develop a sophisticated ability to interpretation the written form in a way that takes into account a myriad of linguistic rules and vague references typical of natural language. Mimicking this ability with semantic processing technology is a difficult challenge.

A semantic processing system must perform tasks that while seemingly simple to a human are not trivial when attempting to codify the rules and data required by a computer system to perform the same task. Added to this is the performance required for semantic information processing of large data sets, most notably requirements for speed and accuracy.

Take for example relatively simple pronoun and anaphora resolutions ("Mike was elected President of the Student Council. If he can do it, so can I"). Before performing any sophisticated natural language analysis like pronoun or anaphora resolution a semantic processing system must first recognize sentence boundaries (not necessarily a simple text parsing task). Then it must parse each sentence into individual tokens (words). After that it must determine the linguistic context of each word and phrase in each sentence (part-of-speech identification, i.e., nouns, noun phrases, verb, etc.). Then finally for resolution of pronouns and anaphora it must determine subject, verb, and object, and then map each abstract subject or object (he, it, etc.) to the concrete form found in earlier sentences. Add to this the need for high-speed processing where applications that search large data sets (mega or tera-byte) must process requests with speeds in the millisecond latency range. And ideally it will take into account the possibility of misspelled words and incorrect punctuation. Then finally, depending upon the situation, it must address word sense disambiguation ("President" in the above example refers to the head of the Student Council, not President of the United States). The net result is that developing a system for semantic processing is not a trivial endeavor.

In addition, a complete system for semantic processing must address all functionality needed to be economically viable, effective and efficient. A notable gap in existing technologies, a gap that has a significant impact on the economic viability of semantic processing, is the lack of automated or semi-automated ontology learning. Ontologies have been recognized as a key part of any semantic processing. For example in the case of the Semantic Web the lack of ontology data is considered a critical stumbling block in realizing the benefits of the Semantic Web [1]. Developing ontologies is a manual process and as such is typically expensive and time consuming. The result is a reduction in the economic viability of semantic processing, i.e., it is neither cost-effective nor timely. Hence the lack of cost-effective ontologies can create a significant stumbling block for any organization seeking a practical semantic processing system.

ROM Estimate of Monetization Potential

While these are rough-order-of-magnitude estimates only, a significant economic potential appears to exist for a successful SPS implementation. A few examples are as follows:

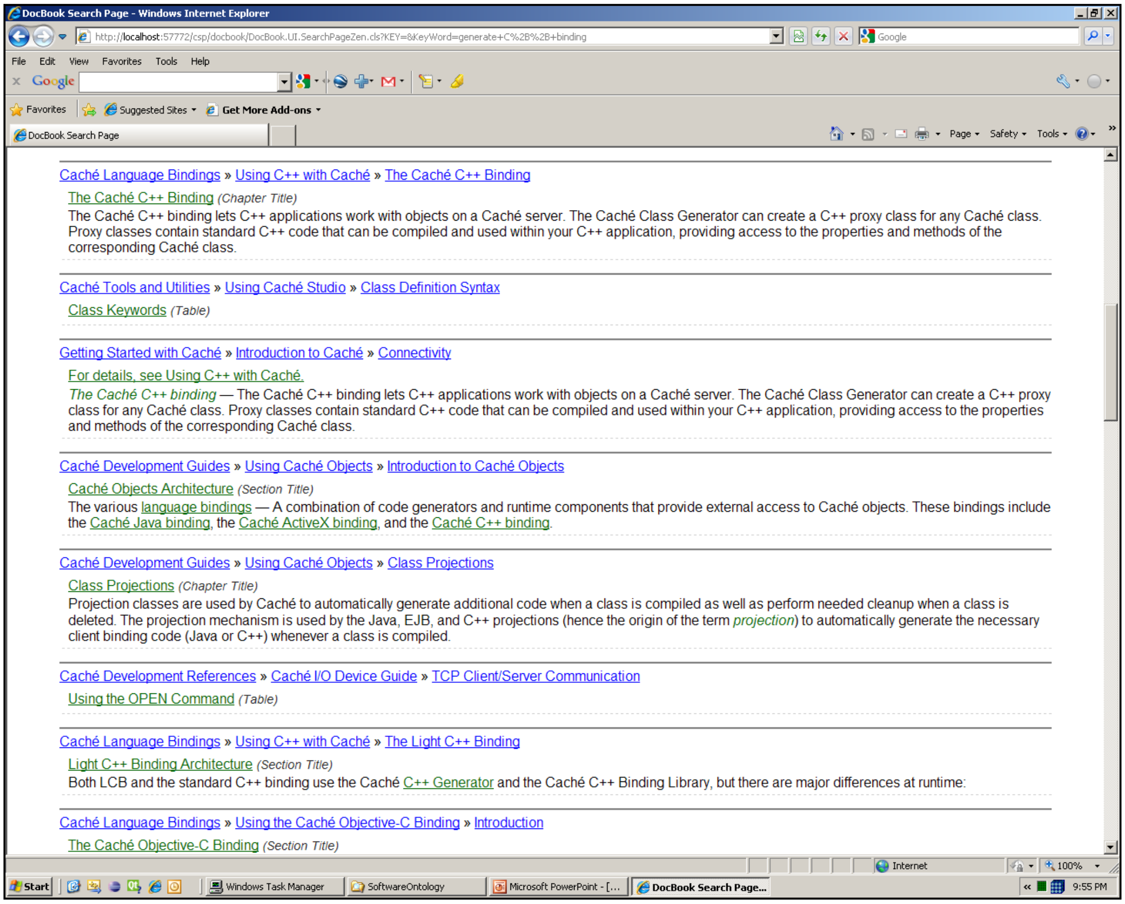
- Research Oriented Search Engines
    - ROM revenue potential for medicine alone: $50MM+
- Knowledge Management
    - Multi-billion dollar industry; ROM revenue potential: $100MM+
- Education Search and Learning Tools, Help-Oriented Search
    - Self-directed learning, software help desk or user self-service, product debugging and maintenance
    - ROM revenue potential: $100MM+

- Military or Other Intelligence Applications
    - ROM revenue potential: $50MM+

The bottom-line is that while a number of serious challenges remain; if these challenges can be overcome the economic benefits appear to exist to make it worthwhile to address these challenges.

Example of Potential Monetization Opportunities

In this example a database engine search tool provides a simple list of results for keywords. The user enters the keywords “generate C++ bindings” and the list below is retrieved.


DocBook Search Page - Windows Internet Explorer
http://localhost:57772/csp/docbook/DocBook.UI.SearchPageZen.cls?KEY=&KeyWord=generate+C%2B%2B+binding
Google
File Edit View Favorites Tools Help
Google
Favorites
Suggested Sites
Get More Add-ons
DocBook Search Page
Page Safety Tools
Internet
100%
Start
Windows Task Manager
SoftwareOntology
Microsoft PowerPoint - [...
DocBook Search Page...
9:55 PM

Caché Language Bindings » Using C++ with Caché » The Caché C++ Binding

The Caché C++ Binding *(Chapter Title)*
The Caché C++ binding lets C++ applications work with objects on a Caché server. The Caché Class Generator can create a C++ proxy class for any Caché class. Proxy classes contain standard C++ code that can be compiled and used within your C++ application, providing access to the properties and methods of the corresponding Caché class.

Caché Tools and Utilities » Using Caché Studio » Class Definition Syntax

Class Keywords *(Table)*

Getting Started with Caché » Introduction to Caché » Connectivity

For details, see Using C++ with Caché.
The Caché C++ binding — The Caché C++ binding lets C++ applications work with objects on a Caché server. The Caché Class Generator can create a C++ proxy class for any Caché class. Proxy classes contain standard C++ code that can be compiled and used within your C++ application, providing access to the properties and methods of the corresponding Caché class.

Caché Development Guides » Using Caché Objects » Introduction to Caché Objects

Caché Objects Architecture *(Section Title)*
The various language bindings — A combination of code generators and runtime components that provide external access to Caché objects. These bindings include the Caché Java binding, the Caché ActiveX binding, and the Caché C++ binding.

Caché Development Guides » Using Caché Objects » Class Projections

Class Projections *(Chapter Title)*
Projection classes are used by Caché to automatically generate additional code when a class is compiled as well as perform needed cleanup when a class is deleted. The projection mechanism is used by the Java, EJB, and C++ projections (hence the origin of the term *projection*) to automatically generate the necessary client binding code (Java or C++) whenever a class is compiled.

Caché Development References » Caché I/O Device Guide » TCP Client/Server Communication

Using the OPEN Command *(Table)*

Caché Language Bindings » Using C++ with Caché » The Light C++ Binding

Light C++ Binding Architecture *(Section Title)*
Both LCB and the standard C++ binding use the Caché C++ Generator and the Caché C++ Binding Library, but there are major differences at runtime:

Caché Language Bindings » Using the Caché Objective-C Binding » Introduction

The Caché Objective-C Binding *(Section Title)*

Figure 1: Example Search Results for Database Engine

An alternative is to provide a learning roadmap-style result. Instead of providing a list of keywords, the user enters the question “How do I generate C++ light bindings?”

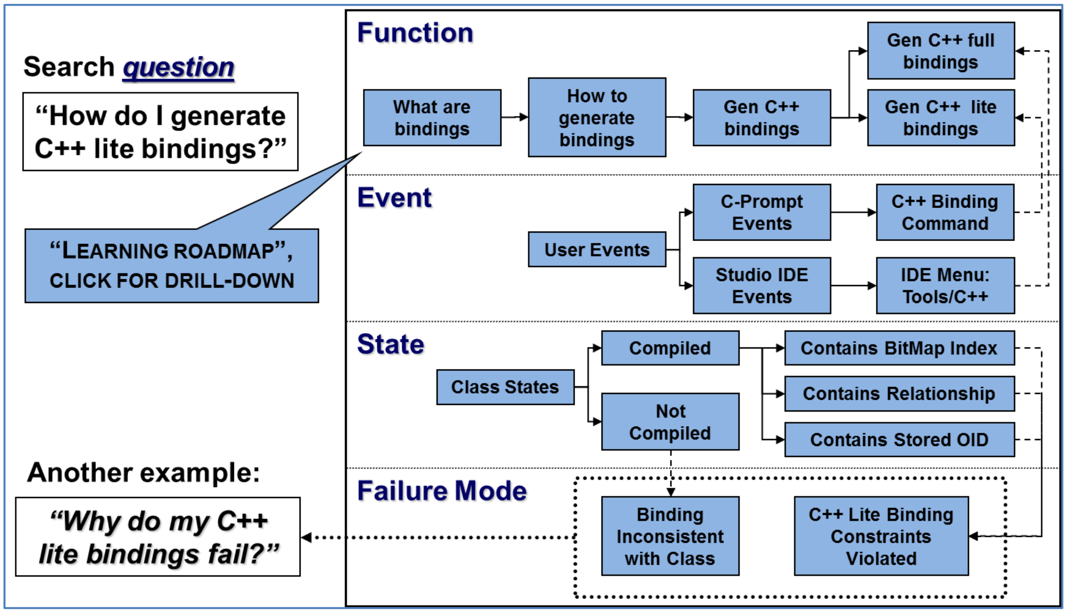


Figure 2: Alternative Database Search - Learning Roadmap

In this example the ontology for the database engine help desk was built around root concepts matching a Failure Modes and Effects Analysis (FMEA) paradigm. In this manner the use could click on subjects (i.e., concepts) to explore more details, either at a more abstract level or detail level, to learn enough about a subject to answer the question posed. The purpose of developing the ontology around the FMEA paradigm was to facilitate answering questions about why something doesn't work, which is in part based upon the product state and events that occurred in addition to product functions. Conceptual feedback from product users was positive, indicating that this approach had merit in terms of having a perceived value to end users.

This is but one example among numerous applications of a practical, real-world implementation of the SPS Framework.

Background

The need for semantic processing has been recognized for quite a few years; hence, a number of technologies exist. A few selected examples, certainly not comprehensive, follows:

- The General Architecture for Text Engineering (GATE) [2], developed at the University of Sheffield, is an open-source Java tool targeting the "development of language processing components". It is roughly equivalent to semantic applications what Eclipse is to Java applications, i.e., an integrated development environment. For example it was used to develop applications for information mining to extract medical events from text [2].
- The Unified Medical Language System (UMLS) [3], created by the US National Library of Medicine, aggregates roughly 100 medical ontologies into one data source for use in information extraction and natural language tools. The NLM also has a variety of tools

for text parsing and part-of-speech tagging [4]. In addition, the NLM also developed MetaMap [5], a tool for natural language processing that parses text and then tags each noun phrase with the matching concept found in UMLS ontologies

- The Never Ending Language Learner (NELL) [6], currently under development at Carnegie-Mellon, is an attempt to mimic how humans learn, that is, using "both context and background knowledge gained over time." NELL is attempting to autonomously learn by reading millions of web pages and extracting/learning stated facts from these data ("Anger is an emotion. Bliss is an emotion."). According to an article published in 2010 by the NY Times [7], NELL has roughly 390,000 learned facts which are purportedly about 87% accurate.
- IBM alphaWorks' LanguageWare [8] is an Eclipse-based application for analyzing unstructured text and extracting facts. It provides a range of functionality such as text parsing, part-of-speech identification, text annotation, and fact mining.

Unresolved Issues for Practical Semantic Processing Systems

While SPS-related technologies are certainly capable in their own right, it was discovered during development of medText that a number of key issues important to practical semantic processing remain unsolved. It is important to note that this prototype included the development of an ontology/topology metric that quantified how well a particular document in the collection being searched matched the search criteria. This added certain requirements for a semantic processing system not apparent in most existing applications. With this addition a number of key issues for a semantic processing system were identified. These include:

1. *Computation of semantic relevancy*:

   MedText implemented a form of search called semantic search, also called "concept-based" search. Just as the name implies, concept-based search uses a mechanism for retrieving relevant documents based upon concepts that exist in the ontology, not Boolean key word logic using the terms provided by a user (aka, "Google" search).

   A standard approach for the quantification of relevancy for concept-based search was not found in the literature prior to the development of medText. One of the main objectives of the medText prototype was to develop and test such an approach.

   When development of medText was almost complete a recently published literature source addressing concept-based search was identified [9]. Test results published in this book supports superior search precision using concept-based search, which was subsequently confirmed on a preliminary basis when performing limited testing of medText.

   What resulted in medText was the calculation of a topological covering space for both the search criteria and each document to be searched. Relevancy for each document is computed by the intersection of these two covering spaces, quantified by the relative size of the intersection compared to the size of the search criteria covering space.

Modeling knowledge context as a topological covering space is reasonable since, due to ontological subsumption, ontologies can be modeled as directed acyclic graphs (DAG). Further details are provided below.

2. *High-speed processing*:
   The calculation of semantic relevancy involves graph computations for the ontological DAG to determine which concepts are related to the set of concepts associated with the search criteria. First it computes the covering space for the search criteria by extracting a sub-graph of the ontological DAG where all ancestors for the search criteria are included in the sub-graph. Then it performs a similar process for each candidate document in the corpus being searched.

   Two factors have a significant impact on computational speed:

   a. Avoiding costly covering space calculations for a candidate document requires the determination of whether or not the covering space for the candidate contains any of the ancestors of the search criteria. Semantic tagging of a document identifies the concepts that exist in a document, but not their ancestors. Hence when scoring a document it is necessary to very quickly compute ancestors on a real-time basis, which has a significant impact on application throughput and latency (i.e., impacts broad market acceptance).
   b. Calculating relevance for a candidate document requires quantification of the size of the intersection between the covering spaces of the search criteria and candidate document (which is itself a covering space). Again, ancestors are not tagged hence this computation is also real-time and hence has a significant impact on throughput.

   Traditional graph computations necessary to perform the above two calculations involve loading the ontological DAG into memory and performing graph walking. However, this is computationally expensive when considering that search performance requirements are typically measured in the millisecond or sub-millisecond range. Further, no industry-standard indexing that accelerates the computation of covering space intersections for arbitrary graphs was found in the literature.

3. *Accuracy of semantic (concept) tagging*:
   The medText prototype used the MetaMap [5] open-source tool to tag phrases in medical text with matching ontology concepts found in the UMLS vocabulary. It uses a set of linguistic rules to identify and score potentially matching concepts [10]. Semantic tagging, since it involves natural language processing, is a difficult and complex task and hence is not 100% accurate. Furthermore, the accuracy of semantic relevancy quantification is highly sensitive to the accuracy of concept tags; due to inheritance structures in the ontology even relatively small inaccuracies in tagging can have an impact on semantic relevancy.

Not only is this evident on a deductive basis it was also evident during testing of medText. If purely by chance the tagging mechanism happens to pick a concept that is removed from the true context of the text being tagged, then the calculated covering space used in the relevancy computations can be dramatically different from the actual covering space. Due to ontological subsumption this can be particularly true if picking an incorrect concept deep in the ontological hierarchy, i.e., those closer to leaf concepts. Hence the accuracy of the semantic relevancy quantification is highly dependent upon the accuracy of the tagging mechanism.

4. *Availability of low-cost ontologies*:
   The medical industry was chosen for the medText prototype simply because the medical industry has already developed a large number of freely available ontologies. The UMLS provided the ontological data for medText. As far as the author is aware, all of these ontologies were developed by hand. Testing of medText was limited to a subset of the UMLS called the Systematized Nomenclature of Medicine ontology (SNOMED) [11][69]. This was done to reduce the uncertainty associated with concept similarity across ontologies when aggregated by the UMLS.

   The subset of SNOMED used for testing medText consisted of approximately 310,000 concepts and 1,340,000 relationships. If development of a large ontology of an order of magnitude equal to SNOMED were required for medText it would have taken at least 2-3 more years of development with a cost in the millions of dollars.

   Given the scale of ontologies required to represent a complete knowledge domain, the economic benefits of low-cost ontologies is significant. It can not only improve the value of semantic search tools in general, via a reduced cost of development, it can also make new technologies that rely upon ontological data become economically viable. This is especially true for small technology startups working on a shoe string budget.

5. *"Learning roadmaps" interface for viewing and understanding search results*:
   The medText prototype simply presented search results in a traditional format – a list of links to documents, ranked by relevancy.

   However, it was soon discovered that when a user is in a "learning" mode, this simple ranking is not sufficient. Users are forced to click on each link and review the document content to determine what in the document is relevant to the search criteria. Furthermore, in some cases a user may be learning new information as they progress through the search, so although they may not have entered a particular topic in the search criteria, in a dynamic learning process they end up having to run multiple searches to finally get all of the relevant information they need to both get an answer to a question and also to understand the answer. Getting an answer and understanding the answer can be two different things that are dynamically mixed together as part of a search.

The use of ontologies to enhance search precision (via concept-based search) provides an opportunity to greatly enhance the user search interface since ontologies provide relationships between concepts. These relationships can be leveraged to provide a network-based display that enhances the learning experience. Results that are directly related to the search criteria are displayed as a "central" box. This central box is surrounded by boxes containing results for semantically related concepts of potential interest to the user. Lines between the boxes represent the relationships to the semantically related concepts, sometimes a complex spider web of relationships depending upon the topic. The user can learn by exploring related topics by simply clicking on a box linked to the original search results and obtain additional, related results.

Take for example a complex search that retrieves information about "dorsolumbar spinal fusion with Harrington rod complicated by post-polio syndrome". This phrase refers to a SNOMED medical concept; it exists in the category of surgical procedures, and contains another concept in the category disease processes that is related to the latent effects of polio. A box is displayed for both concepts, along with all related boxed within a certain distance in the ontology DAG. For example among the boxes displayed will be a box about "dorsolumbar" (an anatomy concept), another box about "spinal fusion", and finally a box about "Harrington rod" (a surgical device concept). If the user needs to understand more about Harrington rods they can click on that box to retrieve a subset of documents in the search results that provide more details about that concept. Optionally they can perform another query to retrieve additional information about Harrington rods alone. This provides an interleaving of actions that is represented graphically as related concepts that the user can explore as needed; hence a display incorporating this type of interface is entitled "learning roadmaps".

The primary benefit of this approach is to accelerate end-to-end learning for more complex topics. This is not an appropriate interface for finding a restaurant or movie. It targets more complex or scientific information that requires an understanding of more basic knowledge before a more complex topic can be completely understood. The learning roadmap provides an end-user with a tool that helps them work through this learning process.

Integrated End-to-End Processing System:

Taking into account all of the above five points, while a number of tools do exist, a complete system performing all aspects of semantic processing is not readily available. The lack of an integrated, end-to-end system is the primary motivation for this research. For example, GATE has been available for around 15 years, but it focuses primarily on text processing. Due to a large part with the addition of semantic relevance computations defined earlier, it is important and necessary to add components for ontology development and learning.

In addition it is necessary to integrate these components to ensure performance objectives can be reached. For example in the development of medText the MetaMap concept tagging application was not available other than through Web Service calls to the NLM web site. This was not an adequate interface when processing a very large volume of data and needing 24x7 real-time search capabilities with latency having an order of magnitude in the millisecond range. The volume and latency requirements dictated a solution hosted on resources with low latency/high throughput (e.g., distributed architecture on high-speed platforms connected via high-speed network or backbone).

Hence adding and integrating all key components provides complete end-to-end processing that enables economically viable, fast and effective semantic processing.

Core Components

As a result of these experiences during the development of medText, a semantic processing system framework, or SPS, was drafted in concept that consists of the following components:

Component	Description	Objective
1. Ontology Development	A suite providing automated or semi-automated tools for authoring ontologies and managing ontological data in a way that dramatically reduces labor requirements.	Low-cost ontologies
2. Natural Language Processing	Parsing of search criteria and candidate documents and then tagging these with ontology concepts.	Speed and throughput appropriate for low latency and big data applications
3. Semantic Relevancy	Index ontological data and perform covering space calculations.	
4. Search and “Learning” User Interface	Graph-based user interface for performing search and learning.	Accelerated learning by end users

Note that the SPS Framework targets ontology-based semantic processing. If non-ontology-based processing is performed, modifications are required or a different framework must be developed.

Core Software Components

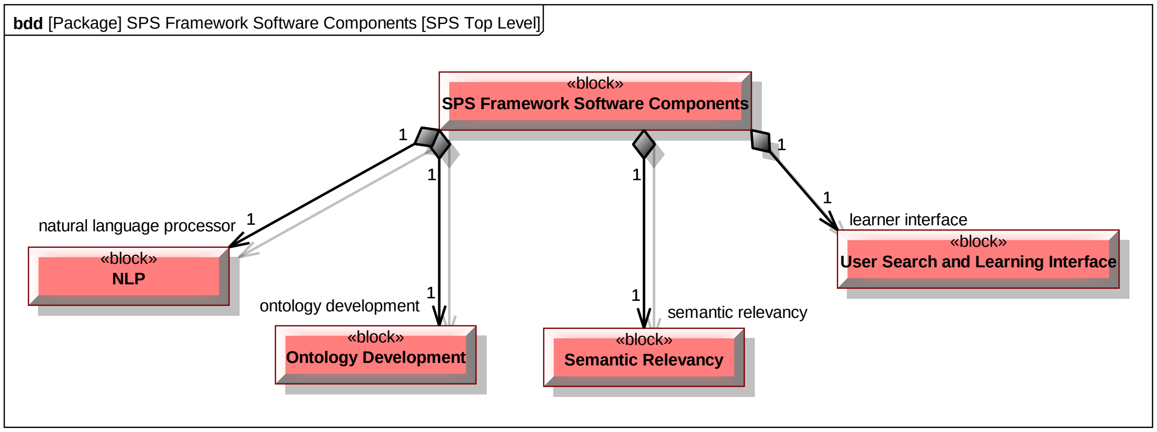
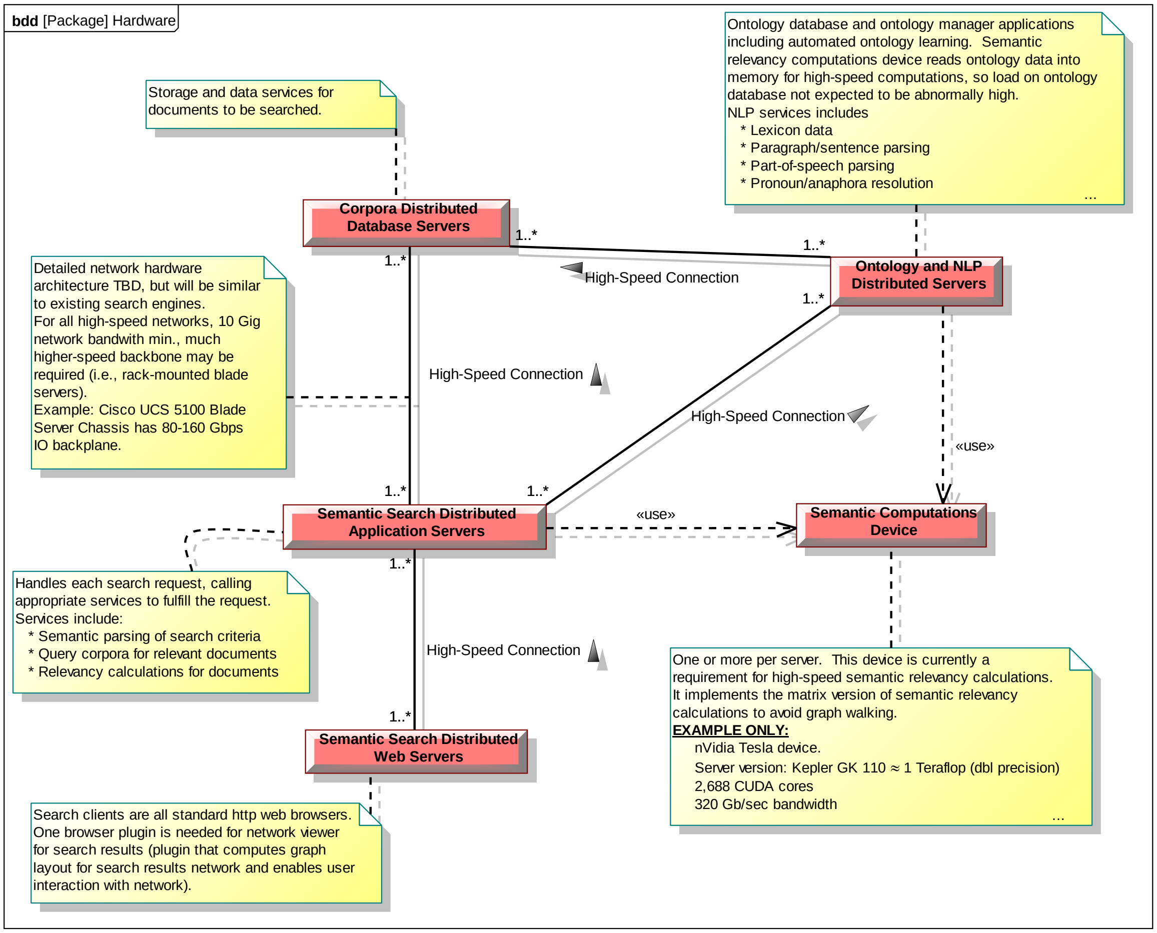

bdd [Package] SPS Framework Software Components [SPS Top Level]
«block»
SPS Framework Software Components
1
1
1
1
1
natural language processor
1
«block»
NLP
ontology development
1
«block»
Ontology Development
1
semantic relevancy
«block»
Semantic Relevancy
learner interface
1
«block»
User Search and Learning Interface

Figure 3: SPS Core Software Components

Core Hardware Components



Storage and data services for documents to be searched.

Ontology database and ontology manager applications including automated ontology learning. Semantic relevancy computations device reads ontology data into memory for high-speed computations, so load on ontology database not expected to be abnormally high.
NLP services includes
* Lexicon data
* Paragraph/sentence parsing
* Part-of-speech parsing
* Pronoun/anaphora resolution
...

Detailed network hardware architecture TBD, but will be similar to existing search engines.
For all high-speed networks, 10 Gig network bandwith min., much higher-speed backbone may be required (i.e., rack-mounted blade servers).
Example: Cisco UCS 5100 Blade Server Chassis has 80-160 Gbps IO backplane.

Handles each search request, calling appropriate services to fulfill the request.
Services include:
* Semantic parsing of search criteria
* Query corpora for relevant documents
* Relevancy calculations for documents

One or more per server. This device is currently a requirement for high-speed semantic relevancy calculations. It implements the matrix version of semantic relevancy calculations to avoid graph walking.
EXAMPLE ONLY:
nVidia Tesla device.
Server version: Kepler GK 110 ≈ 1 Teraflop (dbl precision)
2,688 CUDA cores
320 Gb/sec bandwidth
...

Search clients are all standard http web browsers. One browser plugin is needed for network viewer for search results (plugin that computes graph layout for search results network and enables user interaction with network).

Figure 4: SPS Core Hardware Components

The core hardware components, shown in Figure 4 above, are anticipated for typical distributed systems architecture, similar to those in use today. However, each semantic search application implementation must follow an architecture that is designed and sized to meet the anticipated number of requests workload, which may vary significantly between different semantic processing projects.

All servers are Linux 64-bit, with Intel or AMD CPUs. All applications, with the exception of those required for the nVidia CUDA devices (C++) are written in Java and J2EE.

Web Servers

Search interfaces are performed via a standard web browser, so web servers are required to provide http web interfaces for users to perform queries. The web interface is implemented by the User Search and Learning Interface component shown in Figure 2. The web servers each instantiate multiple threads in a pool sized to service the query demand rate. The web server invokes a query search request to service each request received from an instance of the http connection for the User Search and Learning Interface browser page. This query search request is submitted to the distributed application server bank hosting the semantic search engine front-end application.

Semantic Search Distributed Application Servers

The semantic search application hosted on distributed application servers are the brokers that handle each search request submitted by a user (i.e., submitted via each http instance in the web server thread pools). It will follow a simple, automated work for to fulfill each service request. The functionality for this application is as follows (shown in the same order as the work flow):

1. Submit requests to the ontology and NLP servers to perform the semantic parsing of search criteria submitted by user. Search criteria are received as a simple string of text. Text parsing includes the following:
    a) Parse text into paragraphs, sentences, words, and phrases.
    b) Tag each phrase with a part-of-speech category (noun phrase, verb phrase, etc.).
    c) Tag each phrase with a matching concept found in the ontology.
2. Query corpora database to obtain relevant documents.
3. Perform relevancy calculations for each document.
4. Categorize documents by concepts in the ontology to facilitate user exploration via a “learning roadmap” paradigm. Categorization consists of flagging each concept found in the sub-graph of the ontology to the document if that concept is also found in the document covering space. The user interface for search and learning uses these flags to provide matching documents when a user is walking the learning roadmap and clicks on a concept to follow a learning thread.

None of the above tasks are performed by applications on the semantic search server. The semantic search server is the broker; it forwards these requests to the appropriate server hosting that application.

While performing the above steps the semantic search application will perform load balancing across the distributed servers hosting the ontology and natural language application and the servers hosting the distributed corpora database.

Ontology and Natural Language Processing Distributed Application Servers

These servers host the following processing functions:

a) Store the ontology data and fulfill requests for ontology data,
b) Perform natural language processing, including parsing text into paragraphs, sentences, words and phrases, part of speech tagging, and concept tagging, and pronoun/anaphora resolution
c) Store the lexicon database required to complete NLP requests.

Ontology data storage does not require extensive bandwidth since the ontology will be loaded into memory of GPU devices installed in the servers for ontology calculations.

Ontology development is performed on distributed servers in the same manner as semantic search requests. The browser interface is the main difference. For automated ontology development a browser plugin is required so that the browser becomes a “thick client”, almost to the same level as a typical desk-top application. This is required due to the added functionality required for ontology development. An implementation of the SPS framework may use a client-server model instead of web model whereas the service requests are still via http, but the requests are for web services and the interface is a standard windows-based thick client instead of browser-based.

Semantic Computations Device

This component is a massively parallel computation device that is programmed to perform graph computations for the ontology with extremely low latency. Both the ontology and NLP distributed servers and the semantic search distributed servers will have one or more of these devices installed (multiple devices for high service volume needs, i.e., high bandwidth requirements).

These devices will be nVidia GPU units capable of processing in the 1 teraflop range. These devices are programmed using CUDA/C++. An example configuration is as follows:

nVidia Tesla device
Server version: Kepler GK 110 (≈ 1 teraflop processing speed @ double precision)
2, 688 CUDA cores
320 Gb/s bandwidth

Corpora Distributed Database Servers

This architecture employs distributed database storing the corpora on multiple servers to achieve required high bandwidth. It will use a traditional RDBMS system (e.g., Oracle, MySQL, etc.) that provides distributed database functionality. Inter-server communication between database instances will make the data appear as one database, even though portions are distributed across multiple servers. Communication protocol will be in accordance with the requirements of the database technology. This will be TCP/IP with another layer on top of the TCP/IP for inter-server queries. For example, Oracle provides the Net8 protocol that accepts queries and performs the distributed processing, over TCP/IP network, in a manner that is transparent to the user.

Rack-Mounted Installation, Between Server Communications

Servers are connected using TCP/IP, 10 gig networks, with servers at the same site being mounted on racks with a very high-speed backplane. Unless otherwise noted all inter-process service requests are via RPC calls using Java J2EE applications.

Ontology Data

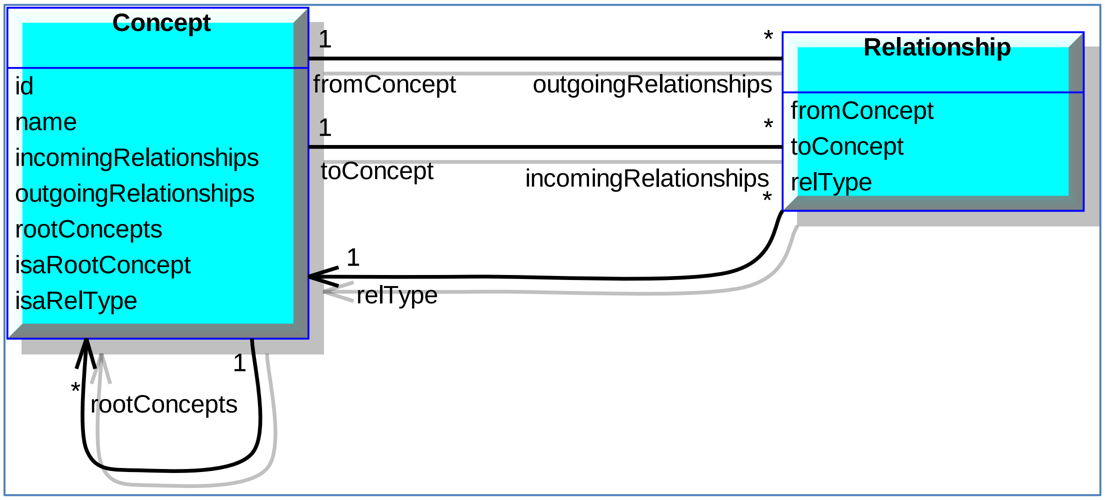


Figure 3: Basic Ontology Persistent Data Structure

As can be seen in Figure 5 the ontology data is conceptually simple since it consists of only two basic classes: a) concepts, and b) relationships. For the medText prototype the SNOMED ontology was stored and accessed using this structure.

The ontology will be loaded into memory in the nVidia GPU devices, so the ontology database is accessed only when the servers are booted. Hence a low latency, distributed approach is not required for the database.

Corpora and Search Criteria Data

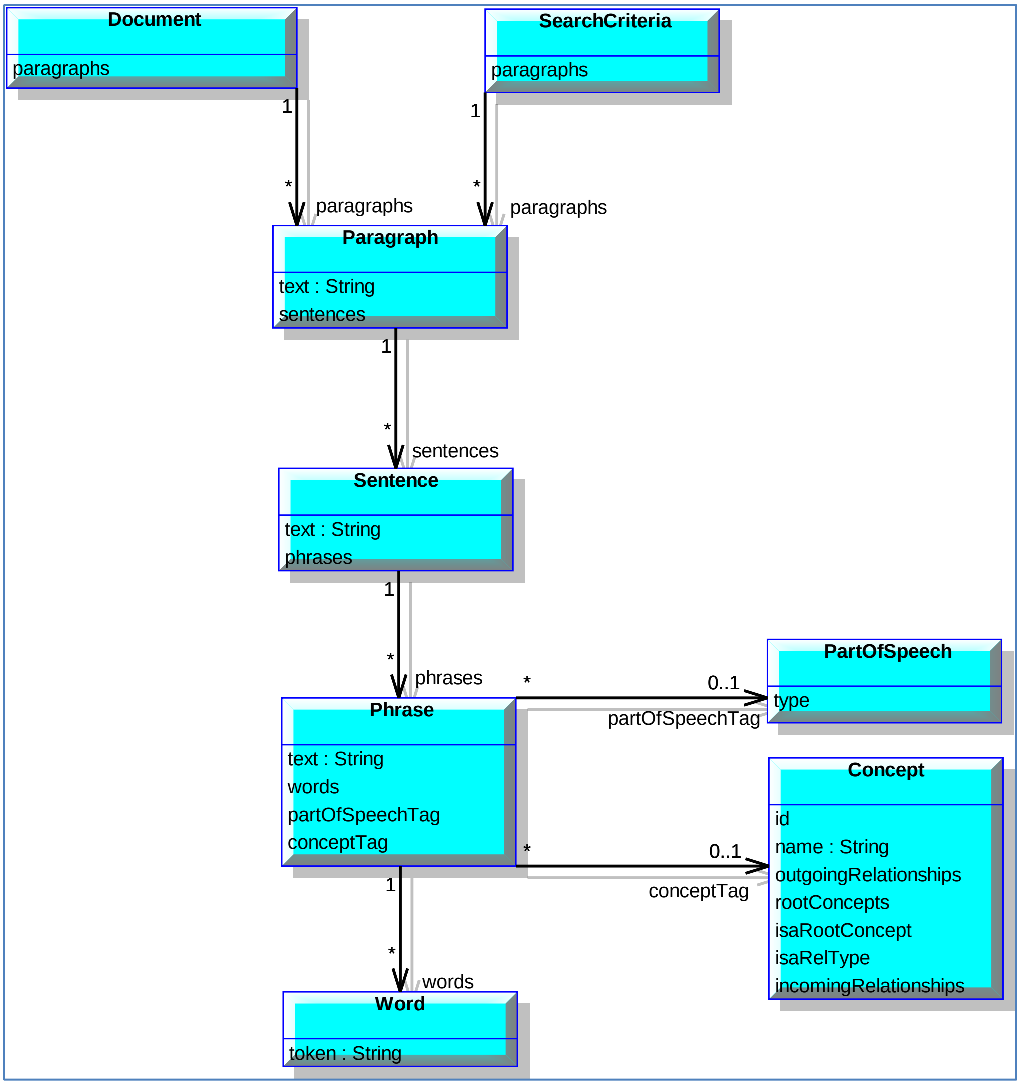


Figure 6: Corpora and Search Criteria, Text, Paragraphs, Sentences, Phrases, and Tags

The corpora data structure consists of the document itself, which consists of paragraphs, phrases, and words. Phrases are tagged with part of speech category and matching concept found in the ontology.

The search criteria are treated just like a document in the corpora. This is done because the search criteria are processed in a manner almost identical to a document.

Software Subsystems

Natural Language Processing Software Subsystems

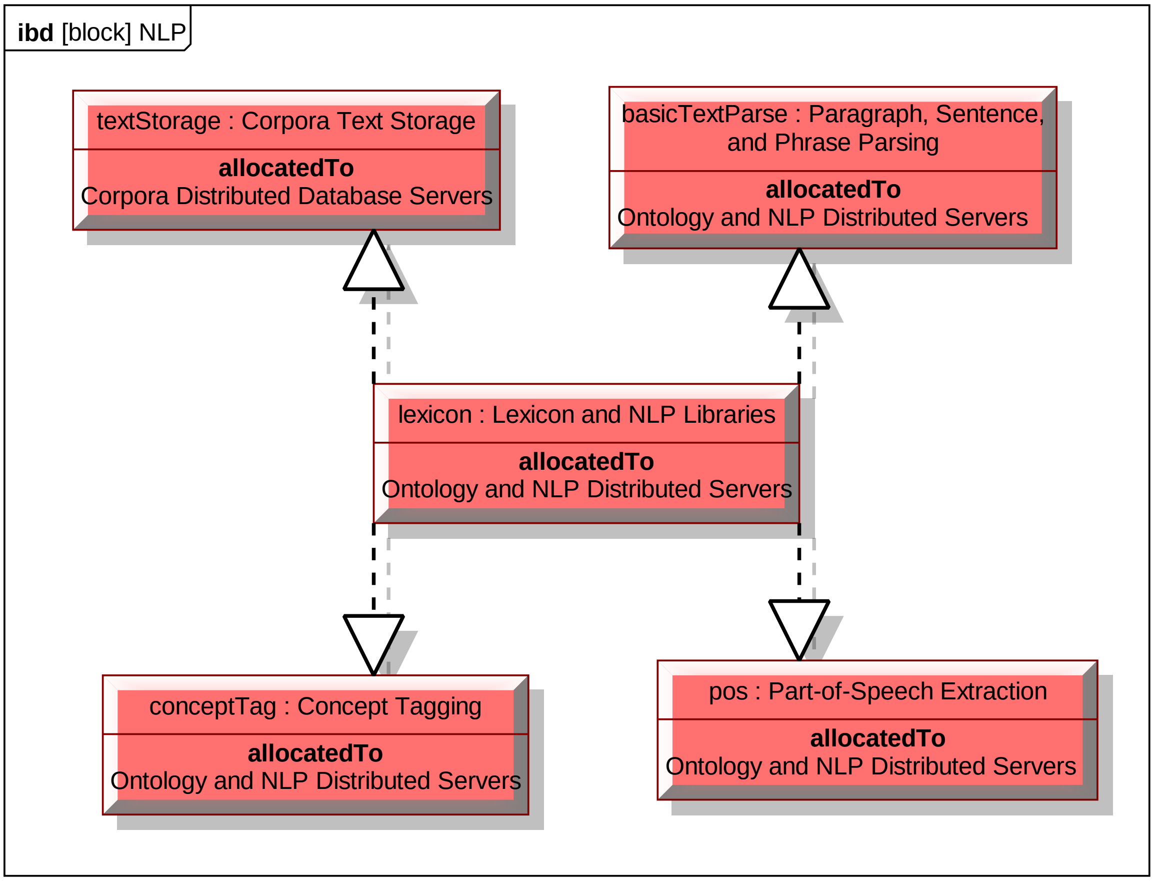

ibd [block] NLP
textStorage : Corpora Text Storage
allocatedTo
Corpora Distributed Database Servers
basicTextParse : Paragraph, Sentence, and Phrase Parsing
allocatedTo
Ontology and NLP Distributed Servers
lexicon : Lexicon and NLP Libraries
allocatedTo
Ontology and NLP Distributed Servers
conceptTag : Concept Tagging
allocatedTo
Ontology and NLP Distributed Servers
pos : Part-of-Speech Extraction
allocatedTo
Ontology and NLP Distributed Servers

Figure 7: Natural Language Processing (NLP) Software Components

The subsystems of the natural language processing software component are shown in Figure 7. The functions associated with these components are as follows:

Component	Description/Function	Host Platform
1. Lexicon and NLP Libraries	Library suite for: a) Storing and retrieving lexicon data b) NLP functions The components described below are the realization of this component.	Ontology and NLP distributed servers.
2. Corpora Text Storage	Storage and indexing of corpora being searched.	Corpora distributed database servers
3. Paragraph, Sentence, and Phrase Parsing	Parsing of text into paragraphs, sentences, words, and phrases. This includes parsing corpora and search criteria (where search criteria are equivalent to a small document).	Ontology and NLP distributed servers

Component	Description/Function	Host Platform
4. Concept Tagging	Perform linguistic or other computations necessary to identify a concept in the ontology that matches a phrase in a sentence.	Ontology and NLP distributed servers
5. Part of Speech Tagging	Uses trained tagged speech components for selection of part-of-speech category for phrases in sentence. Used by concept tagging subsystem.	Ontology and NLP distributed server

Natural Language Processing makes heavy use of lexicons, which is required for syntactic and semantic analysis of phrases and words, e.g., synonyms and morphological variants.

A number of sources exist for these data, such as:

- WordNet
- National Library of Medicine

Insofar as practical implementation of the components for natural language processing, part-of-speech tagging can be stumbling block:

- Training a tagger requires human judgment for a particular domain
- Speed can be improved if POS avoided altogether (potential research topic)

In addition, word sense disambiguation can be stumbling block. This is a common issue with any semantic search tool. Scoping the SPS to a narrow knowledge domain helps to alleviate this concern, but examples were found when testing medText where ambiguous word references did occur.

Ontology Development Software Components

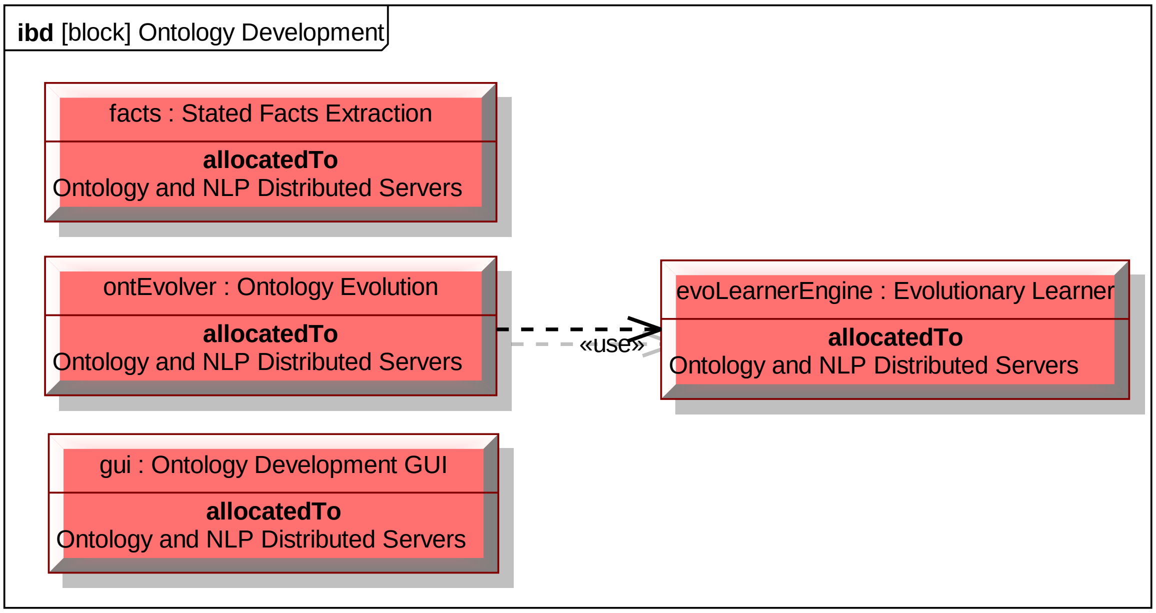

ibd [block] Ontology Development
facts : Stated Facts Extraction
allocatedTo
Ontology and NLP Distributed Servers
ontEvolver : Ontology Evolution
allocatedTo
Ontology and NLP Distributed Servers
gui : Ontology Development GUI
allocatedTo
Ontology and NLP Distributed Servers
evoLearnerEngine : Evolutionary Learner
allocatedTo
Ontology and NLP Distributed Servers
«use»

Figure 8: Ontology Development Software Components

The ontology development software component is hosted on the ontology and NLP distributed servers. It consists of the following:

Component	Description/Function	Host Platform
1. Stated Facts Extraction	Perform lexical analysis and identify facts stated in sentences.	Ontology and NLP distributed servers
2. Ontology Evolution	Perform ontology learning using an evolutionary-style of approach to incrementally build ontology based upon the stated facts extracted by the Stated Facts Extraction component. As a goal completely automated, but in real-world applications likely is semi-automated. Will likely use evolutionary algorithms supplemented by clustering techniques.	Ontology and NLP distributed servers
3. Evolutionary Learner	Evolutionary algorithms library used by #2.	Ontology and NLP distributed servers.

Component	Description/Function	Host Platform
4. Ontology Development GUI	Front end used by ontology author. Includes functions for loading exemplar text that represents knowledge domain being learned, functions for evaluating and confirming/changing entities extracted, and functions for evaluating and confirming relationships extracted. Also includes graph layout functions to present the entire ontology or ontology sub-graphs to user.	Ontology and NLP distributed servers

Semantic Relevancy Software Components

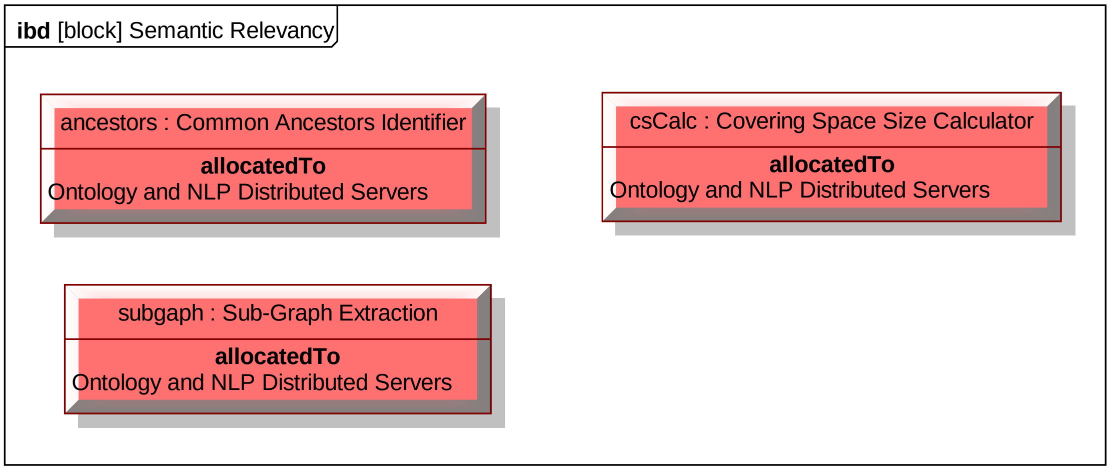

ibd [block] Semantic Relevancy
ancestors : Common Ancestors Identifier
allocatedTo
Ontology and NLP Distributed Servers
csCalc : Covering Space Size Calculator
allocatedTo
Ontology and NLP Distributed Servers
subgraph : Sub-Graph Extraction
allocatedTo
Ontology and NLP Distributed Servers

Figure 9: Semantic Relevancy Software Components

Semantic relevancy consists of first determining if common ancestors exist between the search criteria covering space and the covering space for the document. Then, if common ancestors do exist, a sub-graph for the document is extracted (Sub-Graph Extraction component), and covering space calculations occur (Covering Space Size Calculator component). All components are hosted on the Ontology and NLP distributed servers.

Relevancy calculations are computed by the Semantic Computations Device (nVidia GPU device) installed on the distributed server. These are computed using graph incident matrices as follows:

$$\boldsymbol{incident\ matrix\ A\ of\ graph\ G = n \times n\ matrix\ (i_{ij})}$$
$$\boldsymbol{where\ i_{nj} = 1\ if\ i\ and\ j\ are\ adjacent,\ 0\ otherwise}$$
$$\boldsymbol{A_{covering\ space} = n \times n\ matrix\ (i_{ij})}$$
$$\boldsymbol{where\ i_{nj} = 1\ if\ adjacent\ AND\ in\ covering\ space,\ 0\ otherwise}$$
$$\boldsymbol{For\ CS = CS_a \cap CS_b,\ size_{CS} = (A_a \times A_B)^T \times {I_n}^T}$$

The speed of these calculations is important to achieve internet-level latency typically found in search engines today. For medText the best possible latency for a small document was about 15 milliseconds, but latency in the arena of 1 millisecond is less is necessary. Use of the nVidia devices with 1 teraflop capable throughput appears to make this possible.

Covering Space Calculations and Semantic Relevancy

This section contains additional details regarding the covering space calculations used to determine the semantic relevancy of a document retrieved as part of a search.

All ancestors for one medical concept "Dorsolumbar spinal fusion with Harrington rod" are shown in Figure 10: Example Ontology Snippet: SNOMED-CTFigure 10 below (this concept is a leaf – it has no children). This example is provided to demonstrate that a leaf concept for a complex ontology like SNOMED will likely have a large number of ancestors, 127 in this case. SNOMED has roughly 300,000 concepts and over one million relationships.

Subsumption Relationships in Ontologies

Ontologies have the attribute of subsumption. Subsumption refers to the semantics of broader terms that include all of their descendants. For example, a transportation vehicle has as ancestors the semantics of car, airplane, train, ship, etc. Hence if a document contains a word or a phrase that maps to car, it also includes the concept of transportation vehicle because a car is a transportation vehicle; hence both are in the same topological neighborhood.

The use of subsumption relationships and modeling ontologies as a directed acyclical graph (DAG) are both key to implementing the relevancy calculations described next.

Relevancy Calculation

The covering space, i.e., topological neighborhood, for the concept "Dorsolumbar spinal fusion with Harrington rod" contains the concept itself plus all related concepts in the ontology subsumption hierarchy. Using a graph theory and DAG perspective, this neighborhood consists of all ancestors.

Figure 11 provides an example for calculating relevancy for a fictitious document. The document contains three medical concepts.

Relevancy is the relative size of the intersection of the document covering space with the search criteria covering space. In this case the search criterion contains one concept and the document three.

Taking subsumption into account, the covering space for the documents contains 23 concepts in common with the covering space for the concept "Dorsolumbar spinal fusion with Harrington rod."

Hence the relevancy of this document is 23/127 or 18%.

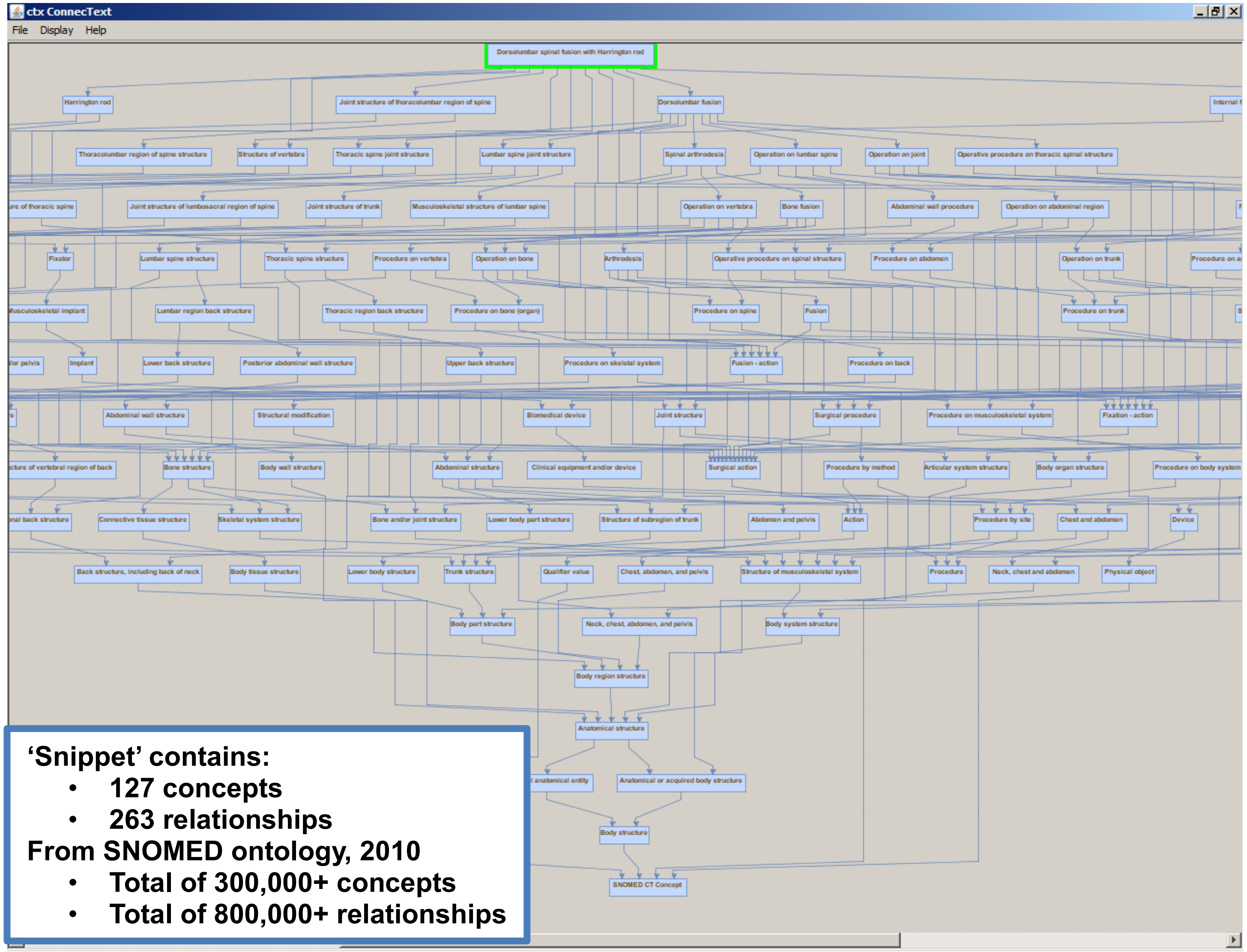
This is a simple calculation but it is mathematically defensible when considering that ontologies provide subsumption relationships, and these define topological neighborhoods. Its validity was verified on a preliminary basis by limited testing of the medText prototype, which suggested superior precision in comparison to the search engine provided by the National Library of Medicine (about 250% higher precision). Due to the limited scope of testing these results are certainly preliminary, but they do suggest that this simple algorithm is worth investigating further.

The stumbling blocks identified in the medText prototype included:

- Speed of relevancy calculations
- Identifying common ancestors to omit non-relevant documents

Graph walking in medText was computationally expensive. The best possible response was in 15 millisecond range for small document (using customized graph walker), which is inadequate for market needs. It does appear that GPU implementations that make use of matrix calculations are capable of required performance.

Example Ontology Snippet from SNOMED-CT



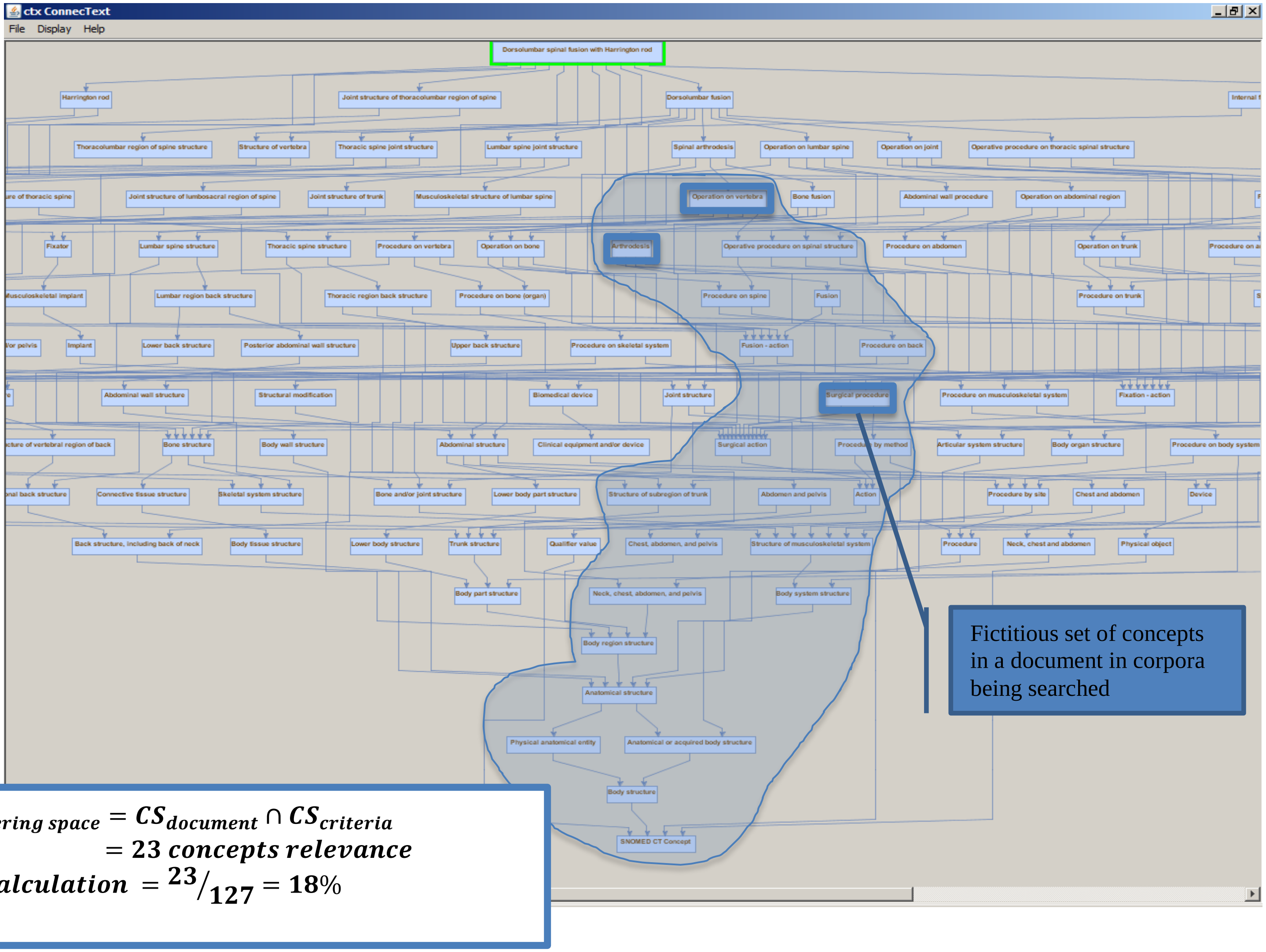
'Snippet' contains:

- **127 concepts**
- **263 relationships**

From SNOMED ontology, 2010

- **Total of 300,000+ concepts**
- **Total of 800,000+ relationships**

Figure 10: Example Ontology Snippet: SNOMED-CT



Fictitious set of concepts in a document in corpora being searched

$$\begin{aligned} relevant_{covering\ space} &= CS_{document} \cap CS_{criteria} \\ &= 23\ concepts\ relevance \\ relevance\ calculation &= {}^{23}/_{127} = 18\% \end{aligned}$$

Figure 11: Relevancy Calculations

II. Use Cases

Use cases described in this section will include additional details regarding the interactions and information flow between SPS software subsystems. This section will provide details on how the software components are linked together to create an integrated system.

Searching and Learning

Use Cases

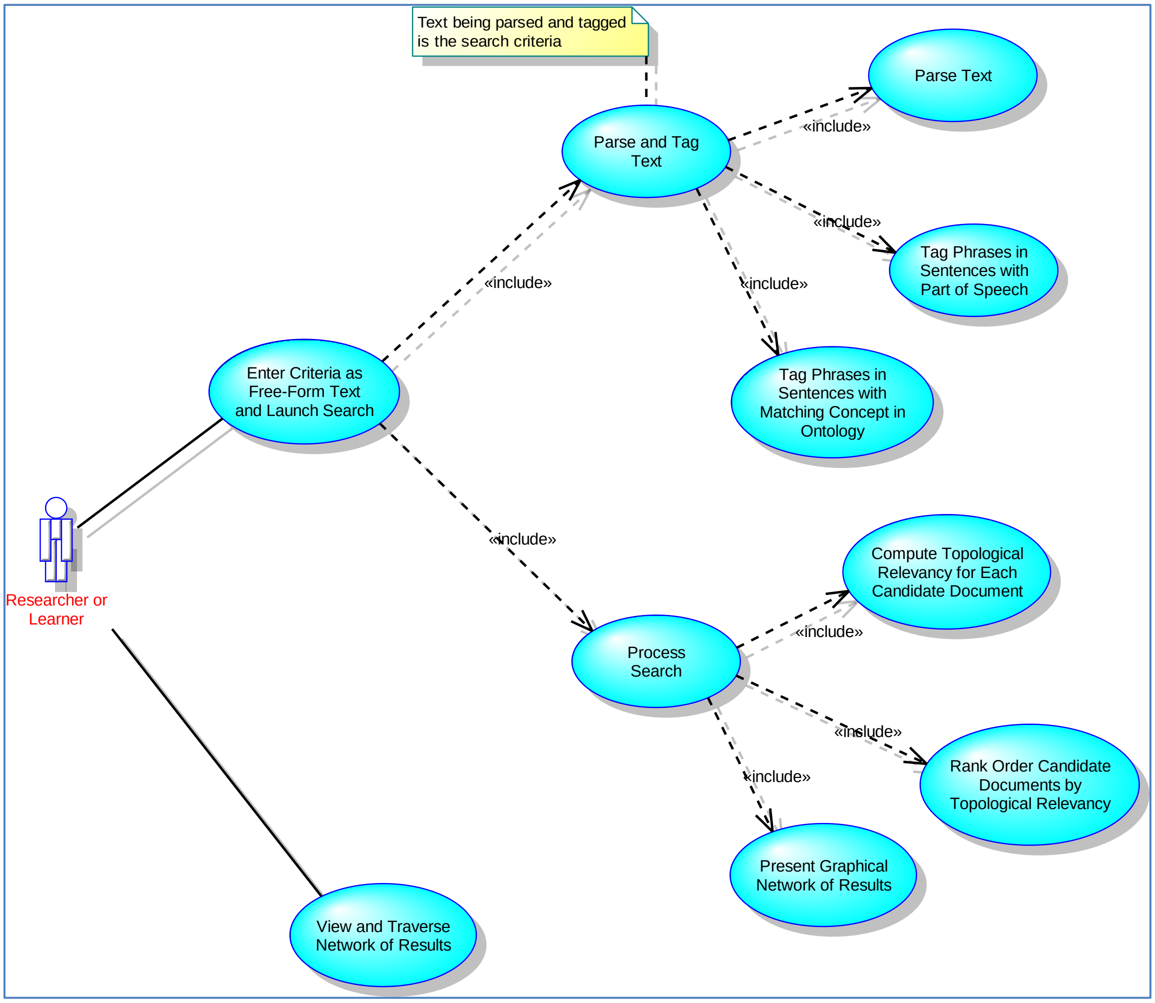

Text being parsed and tagged is the search criteria
Parse and Tag Text
Parse Text
«include»
«include»
Tag Phrases in Sentences with Part of Speech
«include»
Tag Phrases in Sentences with Matching Concept in Ontology
«include»
Enter Criteria as Free-Form Text and Launch Search
Researcher or Learner
«include»
Compute Topological Relevancy for Each Candidate Document
Process Search
«include»
«include»
Rank Order Candidate Documents by Topological Relevancy
«include»
Present Graphical Network of Results
View and Traverse Network of Results

Figure 12: Information Retrieval and Results Use Cases

Overall, Simple User Search Process

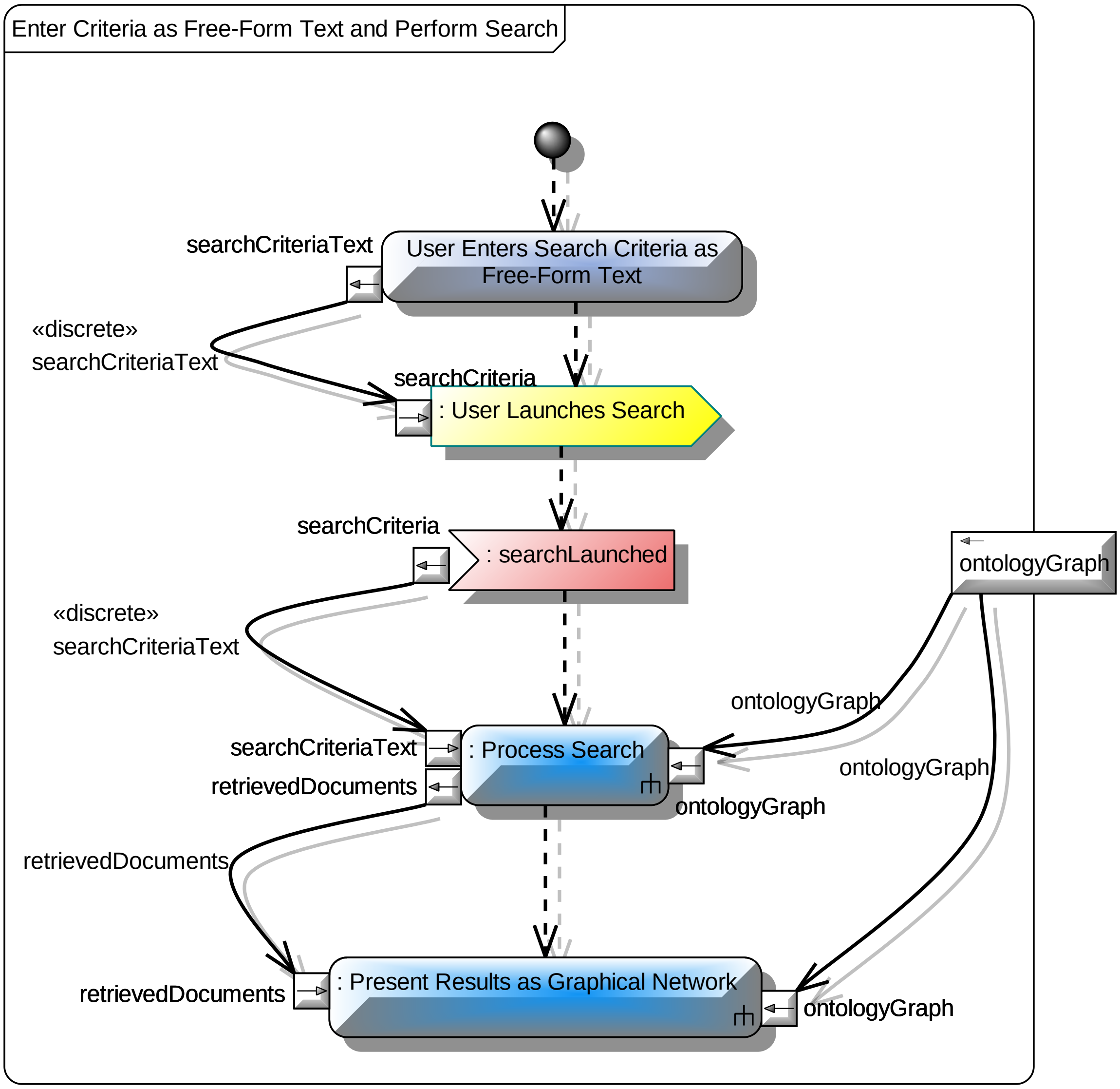


Figure 13: Overall Search Process from User's Perspective

Figure 13 shows a summary of the overall search process. From a user's perspective it is simple. Search criteria are natural language descriptions of a situation or question that user is interested in retrieving information about. The user simply enters the criteria text and clicks on the search button. No keywords or keyword logic is needed.

For medText the search criteria consisted of the history of present illness (HPI) for a small sample of patients (with all identifiers stripped to maintain privacy). Most consisted of 4 or more paragraphs; hence the physician was able to perform complex search using a work product (the HPI) that they create as a normal part of their work activity. This avoided the need for the physician to identify keywords or other information not directly related to a natural language description of the situation, greatly improving ease-of-use.

Process Search

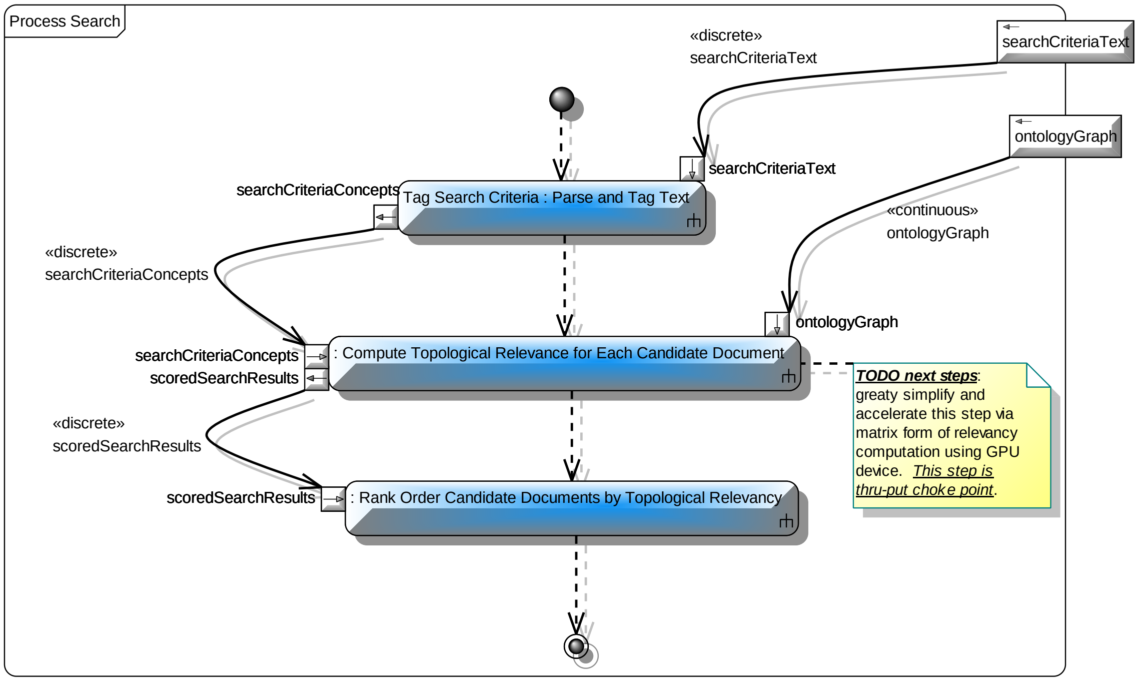

Process Search
«discrete»
searchCriteriaText
searchCriteriaText
ontologyGraph
searchCriteriaText
searchCriteriaConcepts
Tag Search Criteria : Parse and Tag Text
«continuous»
ontologyGraph
«discrete»
searchCriteriaConcepts
ontologyGraph
searchCriteriaConcepts
scoredSearchResults
: Compute Topological Relevance for Each Candidate Document
TODO next steps: greaty simplify and accelerate this step via matrix form of relevancy computation using GPU device. This step is thru-put choke point.
«discrete»
scoredSearchResults
scoredSearchResults
: Rank Order Candidate Documents by Topological Relevancy

Figure 14: Processing Search Request

Processing the search consists of three functions: parsing and tagging the search criteria, computing the topological relevance for each candidate document, and rank ordering results by relevancy.

Parsing and Tagging Search Criteria Text

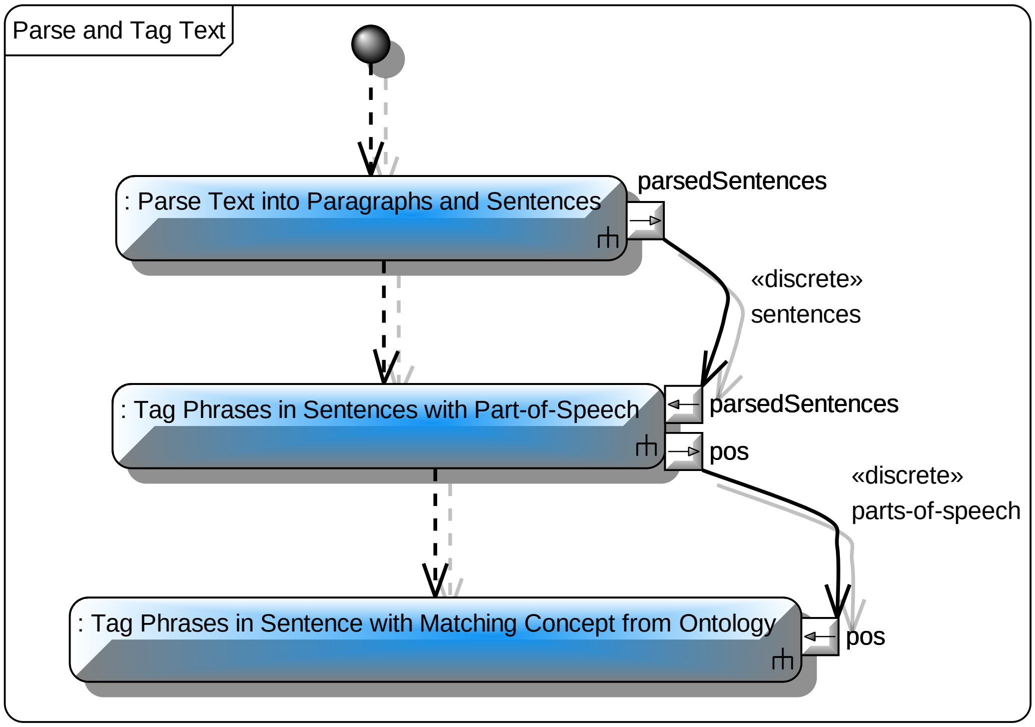

Parse and Tag Text
: Parse Text into Paragraphs and Sentences
parsedSentences
«discrete»
sentences
: Tag Phrases in Sentences with Part-of-Speech
parsedSentences
pos
«discrete»
parts-of-speech
: Tag Phrases in Sentence with Matching Concept from Ontology
pos

Figure 15: Parsing and Tagging Search Criteria Text

The medText prototype used MetaMap from NLM, a free product, to perform these functions. As noted above, MetaMap applies linguistic rules to find best fit concept for noun phrases

However, anecdotal analysis suggested that MetaMap did not appear to achieve a high level of accuracy desired to produce search precision in the 90%+ range.

Use of topological covering spaces under the influence of ontological subsumption appears to drive the sensitivity of search precision to tagging accuracy. If tagging picks the wrong concept, with subsumption this can dramatically change the size of the covering space (i.e., the number of ancestor concepts in common with the search criteria).

The 90% search precision level appears to be required to meet market needs. Discussions with entrepreneurs interested in investing in these technologies were concerned that without a dramatic improvement in accuracy beyond search engines currently available on the market, the probably of market adoption is significantly lower. It appears that search accuracy is highly sensitive to tagging accuracy, and due to user expectations successful market introduction appears to require higher accuracy levels. Hence the accuracy of concept tagging may be a stumbling block to the economic viability of semantic search.

Computing Topological Relevancy (Covering Spaces)

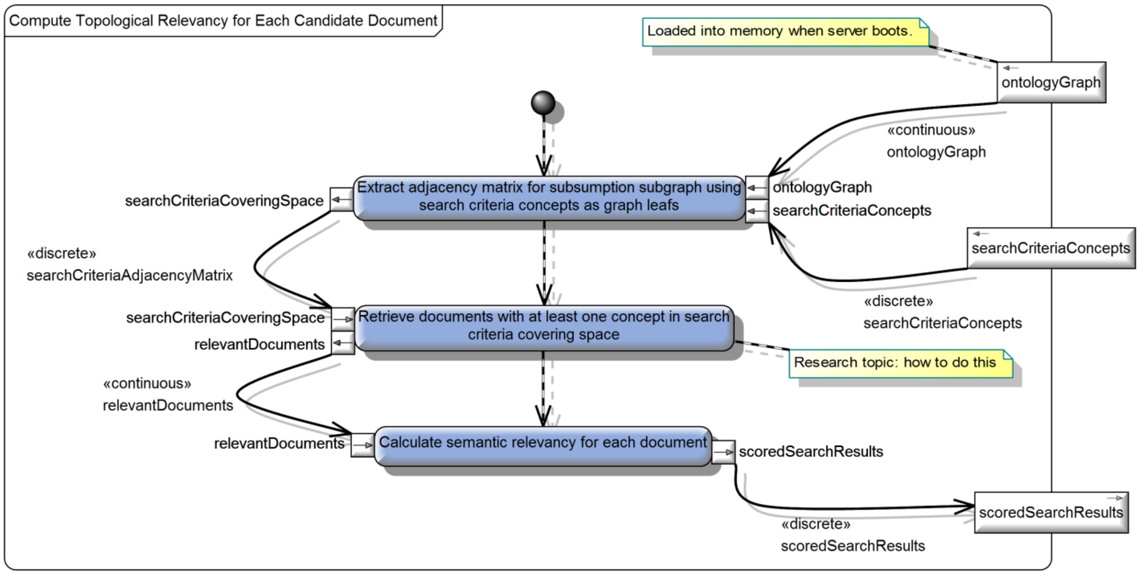

Compute Topological Relevancy for Each Candidate Document
Loaded into memory when server boots.
ontologyGraph
«continuous»
ontologyGraph
Extract adjacency matrix for subsumption subgraph using search criteria concepts as graph leafs
searchCriteriaCoveringSpace
ontologyGraph
searchCriteriaConcepts
searchCriteriaConcepts
«discrete»
searchCriteriaAdjacencyMatrix
«discrete»
searchCriteriaConcepts
searchCriteriaCoveringSpace
relevantDocuments
Retrieve documents with at least one concept in search criteria covering space
Research topic: how to do this
«continuous»
relevantDocuments
relevantDocuments
Calculate semantic relevancy for each document
scoredSearchResults
scoredSearchResults
«discrete»
scoredSearchResults

Figure 16: Computing Semantic Relevancy

Computing semantic relevancy consists of extracting the adjacency matrix for sub-graphs from the ontology graph, retrieving documents with at least one concept in common with the search criteria covering space, and finally calculating semantic relevancy for each document. These steps are associated with the components discussed in the **Covering Space Calculations and Semantic Relevancy** section above.

The key stumbling block to these steps is achieving one or two orders of magnitude improvement in latency (from 15 milliseconds typical in medText to 0.15 milliseconds ±).

Ontology Development via Automated Learning

Use Cases

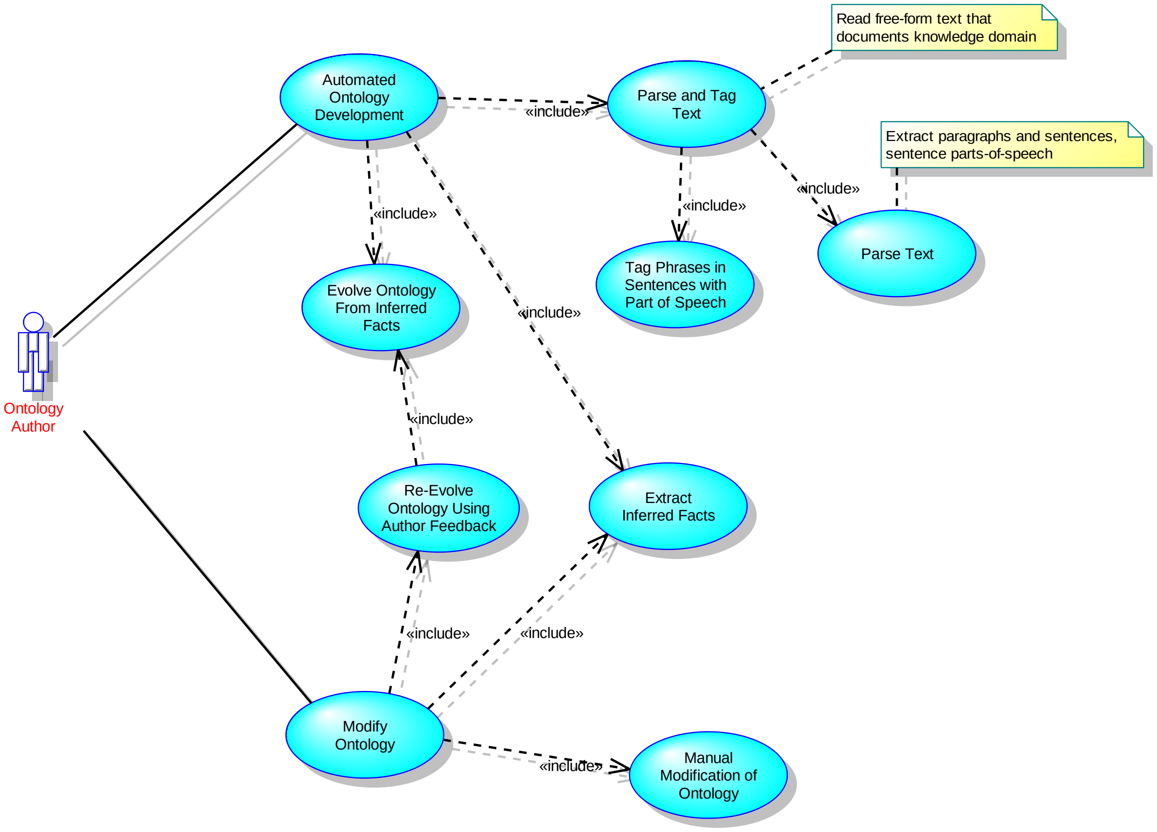

Read free-form text that documents knowledge domain
Extract paragraphs and sentences, sentence parts-of-speech
Automated Ontology Development
Parse and Tag Text
«include»
«include»
«include»
«include»
Tag Phrases in Sentences with Part of Speech
Parse Text
Evolve Ontology From Inferred Facts
«include»
Ontology Author
«include»
Re-Evolve Ontology Using Author Feedback
Extract Inferred Facts
«include»
«include»
Modify Ontology
«include»
Manual Modification of Ontology

Figure 17: Ontology Learning Use Cases

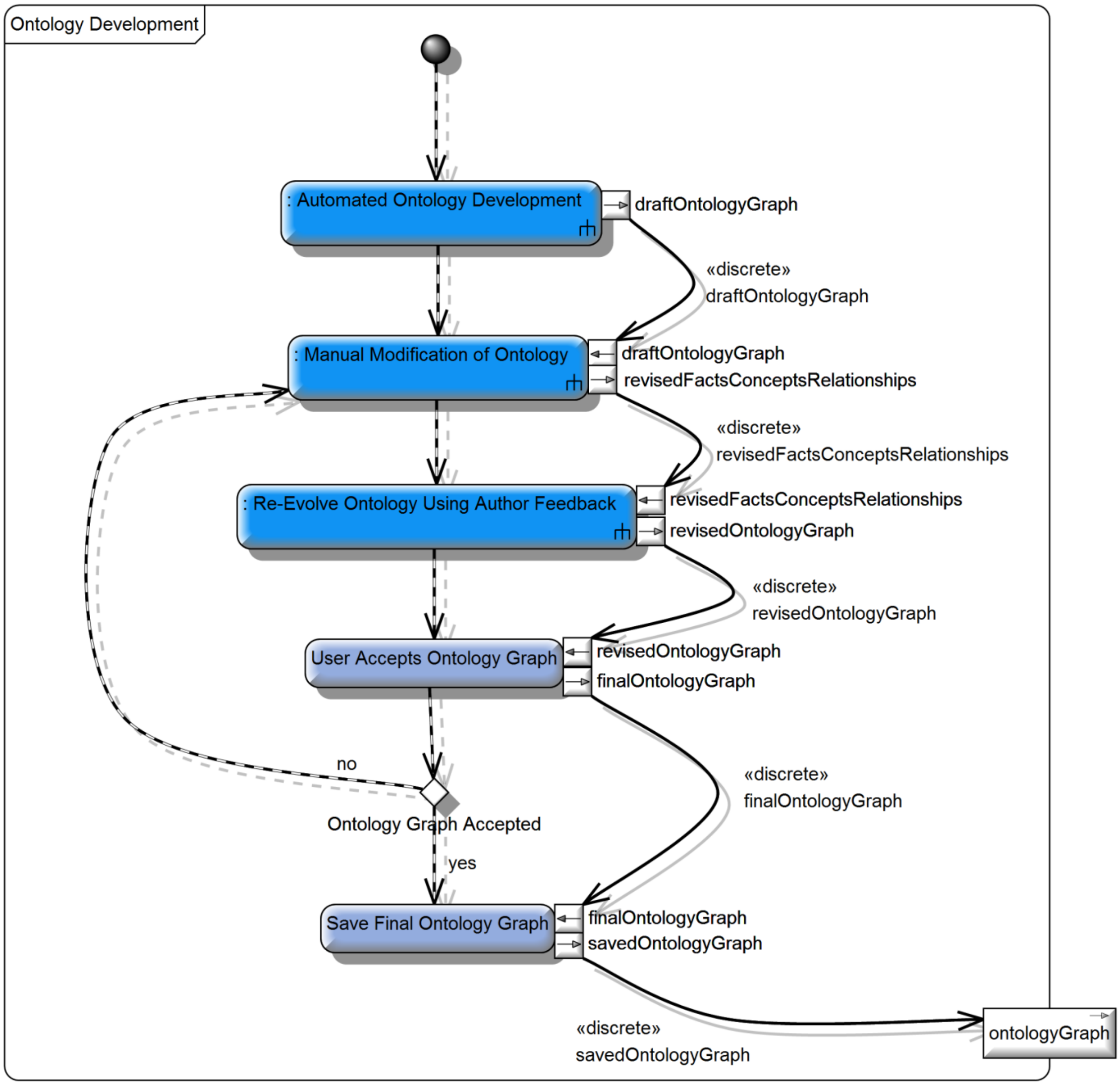


Figure 18: Ontology Development Steps

The objective of these steps is to achieve fully automated ontology development; however, a fully automated approach appears impractical at this time. Hence the approach shown uses person-in-the-loop approach.

The bottom-line goal is to save 80-90% labor over manual approaches.

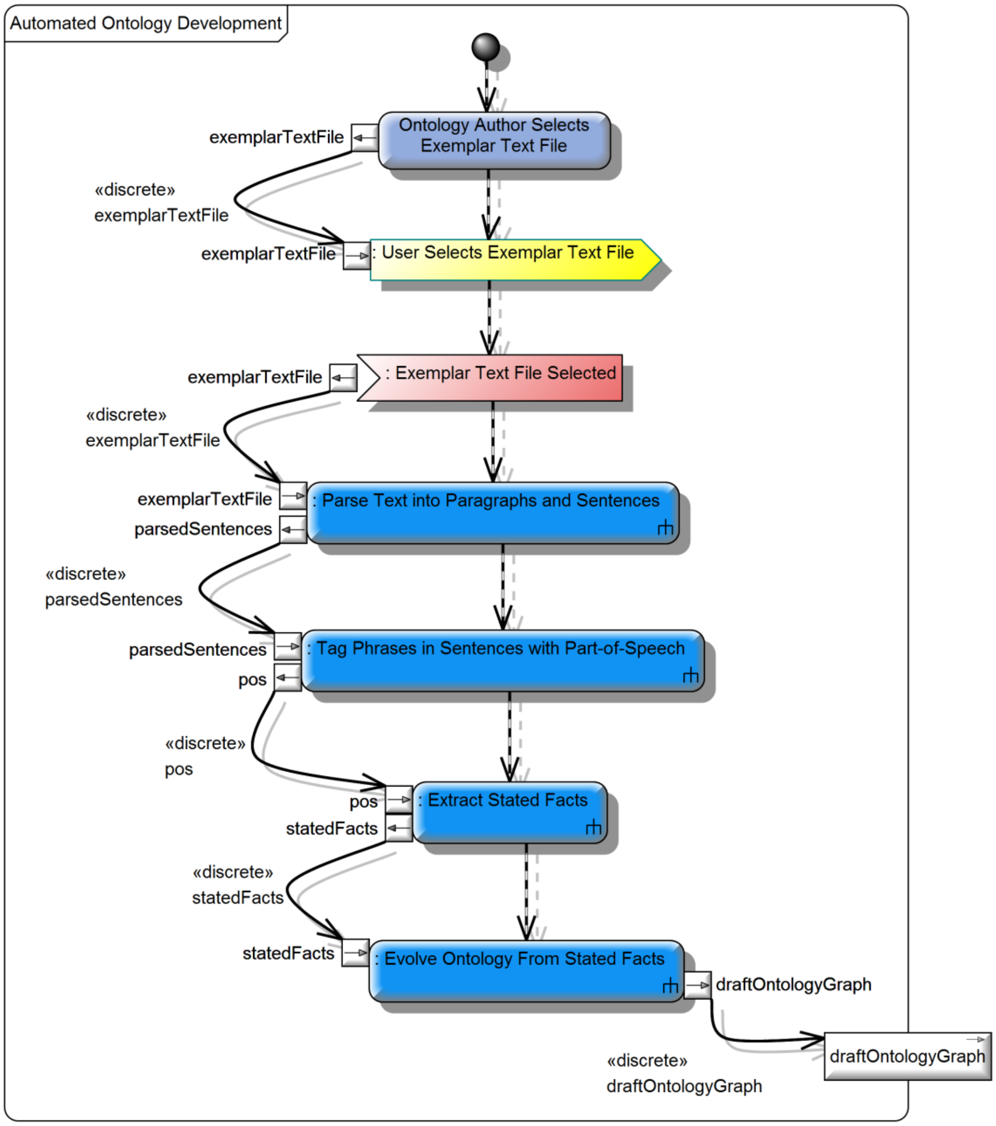

Automated Ontology Development
Ontology Author Selects Exemplar Text File
exemplarTextFile
«discrete» exemplarTextFile
exemplarTextFile
: User Selects Exemplar Text File
exemplarTextFile
: Exemplar Text File Selected
«discrete» exemplarTextFile
exemplarTextFile
parsedSentences
: Parse Text into Paragraphs and Sentences
«discrete» parsedSentences
parsedSentences
pos
: Tag Phrases in Sentences with Part-of-Speech
«discrete» pos
pos
statedFacts
: Extract Stated Facts
«discrete» statedFacts
statedFacts
: Evolve Ontology From Stated Facts
draftOntologyGraph
«discrete» draftOntologyGraph
draftOntologyGraph

Figure 19: Automated Portion of Ontology Development

Figure 19 above shows the process for the automated portion of ontology development within the overall process shown in Figure 18.

Based upon experiences gained with medText and with feedback from investors and potential users, the key enabler for a practical implementation of the SPS Framework is having the ability to evolve ontology within the context of person-in-the-loop feedback as the evolutions occur. The optimum combination appears to be a combination of evolutionary algorithm with clustering techniques, although this is purely speculative at this time.

Note that fact extraction already demonstrated by research staff at Boeing, but a key unknown is the identification of a metric quantifying what is a ‘good’ ontology.

Automated ontology learning appears to be a key to the economic viability of a SPS.

Loading and Indexing Corpora to be Searched

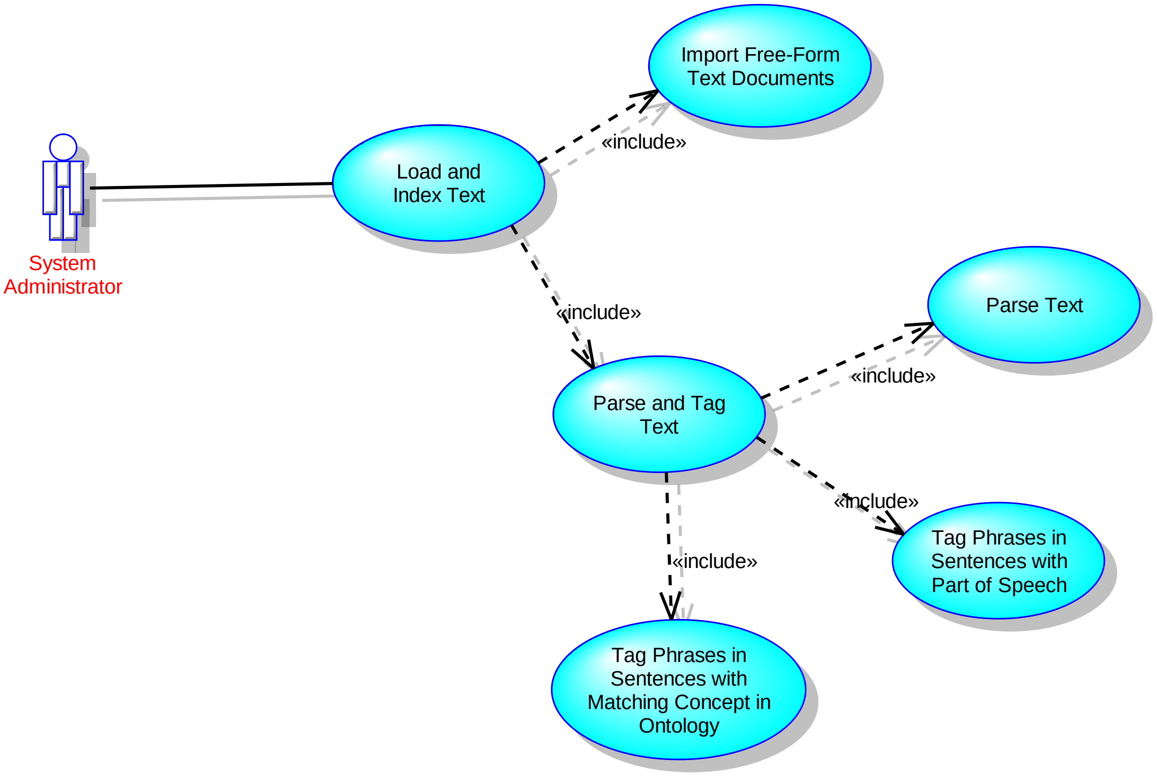

Import Free-Form Text Documents
«include»
Load and Index Text
System Administrator
«include»
Parse Text
«include»
Parse and Tag Text
«include»
Tag Phrases in Sentences with Part of Speech
«include»
Tag Phrases in Sentences with Matching Concept in Ontology

Figure 20: Loading Text into System

As can be seen in earlier sections, loading and indexing corpora to be searched executes all of the functions described earlier.

III. Conclusions

Since many technologies exist today that fill many of the SPS Framework roles, any SPS implementation can reuse existing search engine architectures for distributed systems. This reduces the cost and risk for those investing in new products that make heavy use of SPS technologies.

However, one key technology gap remains – ontology learning – that is key to economic viability and appears to be a significant stumbling block to successful implementation of a practical SPS Framework. For this reason automated ontological learning is the recommended focus for future research.

While other system components require definition, e.g., ontology library management, the majority of SPS functions and components were identified. Based upon these results it appears that the analysis of potential SPS solutions via the SPS Framework has value. It provided a framework that identified key areas that must be addressed to make semantic processing a reality, areas that were consistent with those identified during the development of a prototype SPS system targeting medicine.

IV. References

[1] I. H. T. S. D. Organization, "SNOMED CT," [Online]. Available: http://www.ihtsdo.org/snomed-ct/.

[2] T. U. o. Sheffield, "GATE," [Online]. Available: http://gate.ac.uk/.

[3] IBM, "IBM LanguageWare Resource WorkbenchJoin this Group," [Online]. Available: https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=6adead21-9991-44f6-bdbb-baf0d2e8a673.

[4] N. Shadbolt, W. Hall and T. Berners-Lee, "The Semantic Web Revisited," IEEE Intelligent Systems, pp. 96-101, 2006.

[5] N. L. o. Medicine, "The SPECIALIST NLP Tools," [Online]. Available: http://lexsrv3.nlm.nih.gov/Specialist/Home/index.html.

[6] N. L. o. Medicine, "MetaMap Portal," [Online]. Available: http://metamap.nlm.nih.gov/.

[7] A. Aronson, "Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program," in AMIA 2001 Annual Symposium, Washington, DC, 2001.

[8] A. Schenker, H. Bunke, M. Last and A. Kandel, Graph-Theoretic Techniques for Web Content Mining, Singapore: World Scientific Publishing Co. Pte. Ltd., 2005.

[9] N. L. o. Medicine, "Unified Medical Language System," 2009. [Online]. Available: http://www.nlm.nih.gov/research/umls/.

[10] T. N. Y. Times, "Smarter Than You Think: Aiming to Learn as We Do, a Machine Teaches Itself," 2010. [Online]. Available: http://www.nytimes.com/2010/10/05/science/05compute.html?_r=3&src=twt&twt=nytimesscience&.

[11] C. M. University, "NELL - The Computer that Learns," 2010. [Online]. Available: http://www.cmu.edu/homepage/computing/2010/fall/nell-computer-that-learns.shtml.

BIBLIOGRAPHY

1. Hecht-Nielsen, R., *Cogent confabulation.* Neural Networks, 2005. **18**(2): p. 111-115.
2. Hecht-Nielsen, R. *The Mechanism of Thought*. in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. 2006.
3. Hecht-Nielsen, R., *Confabulation Theory The Mechanism of Thought*. 2007, LaJolla, California: Springer.
4. Solari, S., et al., *Confabulation Theory.* Physics of Life Reviews, 2008. **5**(2): p. 106-120.
5. Jong-Hwan, K., et al., *Two-Layered Confabulation Architecture for an Artificial Creature's Behavior Selection.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2008. **38**(6): p. 834-840.
6. Moskovitch, R., et al., *A Comparative Evaluation of Full-text, Concept-based, and Context-sensitive Search.* Journal of the American Medical Informatics Association : JAMIA, 2007. **14**(2): p. 164-174.
7. *Ontology - Wikipedia*. Available from: https://en.wikipedia.org/wiki/Ontology.
8. NLM. *Unified Medical Language System (UMLS)*. 2013; Available from: http://www.nlm.nih.gov/research/umls/quickstart.html.
9. Aronson, A., *The Current State of MetaMap and MMTX.* 2009.
10. Aronson, A.R., *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.* Proceedings / AMIA . Annual Symposium. AMIA Symposium, 2001: p. 17-21.
11. Aronson, A.R., *The effect of textual variation on concept based information retrieval.* Proceedings : a conference of the American Medical Informatics Association / AMIA Annual Fall Symposium. AMIA Fall Symposium, 1996: p. 373-377.
12. Aronson, A.R., *MetaMap: Mapping Text to the UMLS Metathesaurus.* UMLS White Paper, 2006.
13. Aronson, A.R. and F.M. Lang, *An overview of MetaMap: Historical perspective and recent advances.* Journal of the American Medical Informatics Association, 2010. **17**(3): p. 229-236.
14. NLM. *Semantic Navigator, UMLS Terminology Services (UTS)*. 2013; Available from: https://uts.nlm.nih.gov/home.html.
15. Chan, C.W. *Cognitive informatics: a knowledge engineering perspective*. in *Cognitive Informatics, 2002. Proceedings. First IEEE International Conference on*. 2002.

16. Dang Viet, D. and A. Ohnishi. *Improvement of Quality of Software Requirements with Requirements Ontology*. in *Quality Software, 2009. QSIC '09. 9th International Conference on*. 2009.

17. Haibo, H., Z. Lei, and Y. Chunxiao. *Semantic-based requirements analysis and verification*. in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*. 2010.

18. Huang, S.-L., S.-C. Lin, and Y.-C. Chan, *Investigating effectiveness and user acceptance of semantic social tagging for knowledge sharing.* Information Processing & Management, 2012. **48**(4): p. 599-617.

19. Inay, H., O. Kyeong-Jin, and J. Geun-Sik. *Ontology-Driven Visualization System for Semantic Search*. in *Information Science and Applications (ICISA), 2011 International Conference on*. 2011.

20. Innab, N., A. Kayed, and A.S.M. Sajeev. *An ontology for software requirements modelling*. in *Information Science and Technology (ICIST), 2012 International Conference on*. 2012.

21. Jiehan, Z. and R. Dieng-Kuntz. *Manufacturing ontology analysis and design: towards excellent manufacturing*. in *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*. 2004.

22. Johnson, J., M. Henshaw, and H. Dogan, *An incremental hybridisation of heterogeneous case studies to develop an ontology for capability engineering.* Proceedings of the 22nd Annual International Symposium of the International Council of Systems Engineering, 2012.

23. Kaiya, H. and M. Saeki. *Ontology based requirements analysis: lightweight semantic processing approach*. in *Quality Software, 2005. (QSIC 2005). Fifth International Conference on*. 2005.

24. Kaiya, H. and M. Saeki. *Using Domain Ontology as Domain Knowledge for Requirements Elicitation*. in *Requirements Engineering, 14th IEEE International Conference*. 2006.

25. Kossmann, M., et al. *Ontology-driven Requirements Engineering: Building the OntoREM Meta Model*. in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*. 2008.

26. Kossmann, M., et al. *Ontology-driven requirements engineering with reference to the aerospace industry*. in *Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the*. 2009.

27. Kremen, P. and Z. Kouba, *Ontology-Driven Information System Design.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2012. **42**(3): p. 334-344.

28. Lee, S.W. and R.A. Gandhi. *Ontology-based active requirements engineering framework*. in *Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific*. 2005.

29. Li, S. and L. Shi. *Requirements Engineering Based on Domain Ontology*. in *Information Science and Management Engineering (ISME), 2010 International Conference of*. 2010.

30. Kumar, M., N. Ajmeri, and S. Ghaisas, *Towards knowledge assisted agile requirements evolution*, in *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. 2010, ACM: Cape Town, South Africa. p. 16-20.

31. Merrill, G.H. *A practical multi-ontology approach to knowledge exploration*. in *Biotechnology and Bioinformatics, 2004. Proceedings. Technology for Life: North Carolina Symposium on*. 2004.

32. Ramadour, P. and C. Cauvet. *An Ontology-Based Reuse Approach for Information Systems Engineering*. in *Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on*. 2008.

33. Saad, E.W., et al., *Query-based learning for aerospace applications.* Neural Networks, IEEE Transactions on, 2003. **14**(6): p. 1437-1448.

34. Sarder, B. and S. Ferreira. *Developing Systems Engineering Ontologies*. in *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*. 2007.

35. Soylu, A. and P. De Causmaecker. *Merging model driven and ontology driven system development approaches pervasive computing perspective*. in *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*. 2009.

36. Yun, H. *Research on Building Ocean Domain Ontology*. in *Computer Science and Engineering, 2009. WCSE '09. Second International Workshop on*. 2009.

37. Zhuhadar, L., O. Nasraoui, and R. Wyatt. *Visual Ontology-Based Information Retrieval System*. in *Information Visualisation, 2009 13th International Conference*. 2009.

38. Chen, R.-C. and C.-H. Chuang, *Automating construction of a domain ontology using a projective adaptive resonance theory neural network and Bayesian network.* Expert Systems, 2008. **25**(4): p. 414-430.

39. Hourali, M. and G.A. Montazer, *A New Approach for Automating the Ontology Learning Process Using Fuzzy Theory and ART Neural Network.* Journal of Convergence Information Technology, 2011. **6**(10): p. 24-32.

40. Cross, V. and V. Bathija, *Automatic ontology creation using adaptation.* AI EDAM, 2010. **24**(Special Issue 01): p. 127-141.

41. Davalcu, H., et al., *OntoMiner: bootstrapping and populating ontologies from domain-specific Web sites.* Intelligent Systems, IEEE, 2003. **18**(5): p. 24-33.

42. Dongyeop, K., et al. *Automatically learning robot domain ontology from collective knowledge for home service robots*. in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*. 2009.

43. Liu, C., et al., *Convolution Neural Network for Relation Extraction.* Advanced Data Mining and Applications, 2013. **8347**: p. 231-242.

44. Navigli, R., P. Velardi, and A. Gangemi, *Ontology learning and its application to automated terminology translation.* Intelligent Systems, IEEE, 2003. **18**(1): p. 22-31.

45. Navigli, R. and P. Velardi. *LearningWord-Class Lattices for Definition and Hypernym Extraction*. in *48th Annual Meeting of the Association for Computational Linguistics*. 2010. Uppsala, Sweden.

46. Velardi, P., S. Faralli, and R. Navigli, *OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction.* Computational Linguistics, 2012: p. 655-697.

47. Schenker, A., et al., *Graph-Theoretic Techniques for Web Content Mining*. Series in Machine Perception and Artificial Intelligence, ed. H. Bunke and P.S.P. Wang. 2005, Hackensack, NJ: World Scientific Publishing Co. Pte. Ltd.

48. Croft, W. and D.A. Cruse, *Cognitive Linguistics*. 2004: Cambridge University Press. 356.

49. Radden, G. and R. Dirven, *Cognitive English Grammar*. Cognitive Linguistics in Practice, ed. G. Radden. 2007, Amsterdam, The Netherlands: John Benjamins Publishing Company. 374.

50. Pipitone, A. and R. Pirrone. *Cognitive Linguistics as the Underlying Framework for Semantic Annotation*. in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. 2012.

51. Medicine, N.L.o., *MEDLINE citations annotated with disorder mentions*. 2015.

52. Gerstner, W., et al., *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. 2014, Cambridge, United Kingdom: Cambridge University Press. 577.

53. Bastiaansen, M. and P. Hagoort, *Oscillatory neuronal dynamics during language comprehension*, in *Progress in Brain Research*, K. Christa Neuper and Wolfgang, Editor. 2006, Elsevier. p. 179-196.

54. Berners-Lee, T., *WWW: past, present, and future.* Computer, 1996. **29**(10): p. 69-77.

55. Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web.* Scientific American Magazine, 2001(May).

56. Maedche, A. and S. Staab, *Ontology learning for the Semantic Web.* Intelligent Systems, IEEE, 2001. **16**(2): p. 72-79.

57. Chen, R.-C., J.-Y. Liang, and R.-H. Pan, *Using recursive ART network to construction domain ontology based on term frequency and inverse document frequency.* Expert Systems with Applications, 2008. **34**(1): p. 488-501.

58. Fortuna, B., N. Lavrač, and P. Velardi, *Advancing Topic Ontology Learning through Term Extraction*, in *PRICAI 2008: Trends in Artificial Intelligence*, T.-B. Ho and Z.-H. Zhou, Editors. 2008, Springer Berlin Heidelberg. p. 626-635.

59. Gherasim, T., et al., *Methods and Tools for Automatic Construction of Ontologies from Textual Resources: A Framework for Comparison and Its Application*, in *Advances in Knowledge Discovery and Management*, F. Guillet, et al., Editors. 2013, Springer Berlin Heidelberg. p. 177-201.

60. Healy, M.J. and T.P. Caudell. *Generalized Lattices Express Parallel Distributed Concept Learning*. in *Fuzzy Systems, 2006 IEEE International Conference on*. 2006.

61. Zhang, R.-l. and H.-s. Xu. *Using Bayesian Network and Neural Network Constructing Domain Ontology*. in *Computer Science and Information Engineering, 2009 WRI World Congress on*. 2009.

62. Huth, A.G., et al., *Natural speech reveals the semantic maps that tile human cerebral cortex.* Nature, 2016. **532**(7600): p. 453-458.

63. Manning, C.D. and H. Schütze *Foundations of Statistical Natural Language Processing*. 1999, Cambridge, Massachusetts: The MIT Press. 680.

64. Ho, T.K., *Random Decision Forests*, in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. 1995: Montreal, Canada. p. 278-282.

65. Skymind. *Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0*. Available from: http://deeplearning4j.org.

66. Meyer, D., et al., *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. 2015.

67. Kuhn, M., et al., *caret: Classification and Regression Training*. 2016.

68. Shah, N.H., et al., *Comparison of concept recognizers for building the Open Biomedical Annotator.* BMC Bioinformatics, 2009. **10**(Suppl 9): p. S14.

69. Organization, I.H.T.S.D., *SNOMED CT.*

VITA

Dr. Shannon is an experienced leader of 30 years in small and large business.

In 2017 Dr. Shannon received his PhD in Systems Engineering from the Missouri University of Science and Technology (Missouri S&T), with emphasis in computational intelligence. His research included the development of new computational intelligence algorithms for natural language processing. He placed 2nd in the annual Missouri S&T research competition in a field of 50 contestants.

Dr. Shannon earned an M.S. and B.S. in Engineering Management from Missouri S&T in 1984 and 1979.

Dr. Shannon was Founder and President of Raphael Analytics, Inc., leading the development of natural language processing tools for medical text.

As Managing Consultant at Spherion, Dr. Shannon provided IT Quality Assurance services to large IT departments, including IT strategy planning sessions, cross-functional team facilitation, and program management up to the CIO level.

As Assistant to the President/CEO at SCI, Dr. Shannon led a re-engineering team that achieved positive cash flows that enabled doubling the size of the company. Dr. Shannon also held the positions of Director of Customer Satisfaction, and Manager, Property Systems Application Suite.

Dr. Shannon's performed medical outcomes research at Washington University Medical School. He also performed healthcare re-engineering at BJC, introducing new patient, physician, and staff-focused design methods.

As Senior Engineer and Principle Investigator at McDonnell Douglas, Dr. Shannon led the development and application of risk management approaches for a new aerospace vehicle intended to replace the space shuttle.

He is a past Examiner and Senior Examiner for the Missouri Quality Award.

Dr. Shannon is an Eagle Scout, and greatly enjoyed three summers working at the Philmont Scout Ranch in the Sangre de Christo Mountains of New Mexico.