

Question Answering for Biomedicine

by

Yifeng Liu

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science
University of Alberta

© Yifeng Liu, 2016

Abstract

The field of biomedicine is reeling from “information overload”. Indeed, biomedical researchers find it almost impossible to stay current with published literature due to the vast amounts of data being generated and published. As a result, they are turning to text mining. Over the past two decades the field of biomedical text mining has experienced significant advances, such as the development of high quality biomedical knowledge bases and ontologies, the construction of biomedical search engines and the development of biomedical relationship mining tools.

However, users still have to manually examine the retrieved documents and connect snippets of information from various databases to find answers to their queries. Ideally what is needed is a “wise” question answering (QA) system. With the advances in QA systems, including the triumph of IBM Watson on *Jeopardy!*, many biomedical researchers, including myself, believe that now is the time to further advance biomedical text mining by developing a biomedical question answering system. Such a system would be able to answer questions regarding biomedical entities and help researchers better digest existing knowledge and formulate new hypothesis. The task of biomedical question answering is faced with two central challenges: 1) retrieving relevant information from heterogeneous data sources (structured databases and free-text collections), and 2) formulating natural language answers from retrieved concepts and snippets. My research focuses on developing an association mining tool (PolySearch2) and a web-based biomedical question answering system (BioQA), that would provide precise answers with encyclopedia-like commentary to a wide range of biomedical questions. In particular, PolySearch2 mines concept associations from free-text collections based on co-occurrence statistics. BioQA uses PolySearch2 and other tools to decode natural language questions and formulate natural language answers for both descriptive and associative queries. Both

PolySearch2 and BioQA offer public web interface to answer questions posed by biomedical researchers, physicians, students and the inquisitive public. PolySearch2 and BioQA represent an integrated solution to the core challenges in biomedical question answering.

Preface

This thesis is an original work prepared by Yifeng Liu. It arose from ideas suggested by Professor David Wishart at the University of Alberta. The PolySearch2 system referred to in Chapter 3 and the BioQA system referred to in Chapter 4 were designed and implemented by Yifeng Liu, with the assistance of Professor David Wishart. Yifeng Liu was also responsible for conducting the data analysis required to evaluate both systems. Yongjie Liang assisted with the implementation of the PolySearch2 web server. Michael Wilson assisted with setting up and running the ElasticSearch server. Anchi Guo helped with data collection for the biomedical thesaurus. The Chemical Ontology described in Chapter 3 and Chapter 4 is an original work by Yannick Djoumbou. David Arndt and Tanvir Sajed assisted with the administration and maintenance of PolySearch2 and BioQA public web server. The introduction in Chapter 1, the literature reviews in Chapter 2, the algorithms in Chapter 5, and the concluding analysis in Chapter 6 are all original works by Yifeng Liu.

Certain parts of this thesis have been previously published. Chapter 3 expands on the materials published in the journal: *Nucleic Acids Research* under the reference of “Liu, Y., Liang, Y., Wishart, D. (2015) PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Research*. 2015 Jul 1;43(W1):W535-42. doi: 10.1093/nar/gkv383.” on April 29, 2015. Yifeng Liu was responsible for the design, implementation, and evaluation of PolySearch2, as well as the manuscript composition. Yongjie Liang assisted with the implementation of PolySearch2 web server. Professor David Wishart was the supervisory author and was involved with concept formation and manuscript composition.

Acknowledgements

I would like to thank my supervisor Dr. David Wishart for taking me on this fascinating research journey and providing immense help, insightful guidance, and encouragement throughout this journey. Special thanks goes to members of my supervisory and examination committee: Dr. Russell Greiner, Dr. Paul Lu, Dr. Osmar Zaiane, Dr. Duane Szafron, Dr. Warren Gallin, and Dr. Paul Pavlidis for their guidance and assistance.

I would like to express my gratitude to members and staff of the Bioinformatics Research Lab and the Wishart Research Lab, especially Michael Wilson, Yongjie Liang, An Chi Guo, Yannick Djoumbou, Craig Knox, Beomsoo Han, David Arndt, Tanvir Sajed, and Mark Berjanskii for being my source of inspiration over the years. Thanks also goes to my fellow graduate students at the Departments of Computing Science and Biological Science: Jianguo Xia, You Zhou, Noor Hafsa, Fatemeh Miri, He Hua, Zhaochen Guo, Ying Xu, and Hsiu-Chin Lin for countless insightful discussions.

Last but not least I would like to thank my parents, family and friends, who made my life and these studies possible. I want to thank my parents Guoxian Liu and Cuifang Zhan for bringing me into this world and showing me how to be a good person. To my wife Haier and daughter Zoey for lighting up my life with love and joy. I want to thank my extended family, Shu Lun Yu, Yu Mei Yu, Haiyan Yu, and Wenjie Zhong for their love and support. I would also like to thank Dr. Charles Chan Kwok Keung, Dr. Larry Wang, Dr. Jennifer Jay, and Johnson Yeung who have supported and helped me in countless ways.

Funding for this project was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), Canadian Institutes for Health Research (CIHR), Alberta Innovates Health Solutions, Genome Alberta and Genome Canada.

Table of Contents

Abstract	ii
Preface	iv
Acknowledgements	v
Table of Contents	vi
List of Tables	ix
List of Figures	xii
List of Abbreviations	xv
List of My Related Publications	xvi
1. Introduction	1
1.1 Introduction to biomedical question answering	1
1.2 Challenges	4
1.3 Thesis Statement	9
1.4 Thesis Outline	10
2. Background and Related Works	12
2.1 Text Mining Overview	12
2.1.1 Natural Language Processing	12
2.1.2 Machine Learning	18
2.1.3 Information Retrieval.....	19
2.1.4 Information Extraction.....	22
2.1.5 Evaluation Metrics.....	24
2.2 Related Work	28
2.2.1 Biomedical Thesaurus, Lexicons, and Ontologies.....	28
2.2.2 Document Retrieval	29
2.2.3 Named Entity Recognition.....	30
2.2.4 Ontology Matching	31
2.2.5 Relationship Extraction.....	32
2.2.6 Question Answering.....	33
3. PolySerach2: A Text Mining Framework	38
3.1 Introduction	38

3.2 The PolySearch algorithm.....	40
3.3 Improvements and Enhancements in PolySearch2	43
3.3.1 Algorithmic Improvements	43
3.3.2 Graphical Interface and Web Implementation	45
3.3.3 Database and Text Search Enhancements.....	54
3.3.4 Improved Synonym Collections	56
3.3.5 Caching and Auto-Updating	58
3.4 Performance Evaluation.....	58
3.5 Limitations.....	62
3.6 Conclusion	63
4. BioQA: An Automated Biomedical Question Answering System	64
4.1 Introduction.....	64
4.2 BioQA’s User Interface	67
4.3 BioQA’s Knowledge base	78
4.4 BioQA’s Algorithms.....	85
4.5 Performance Evaluations	98
4.5.1 Question Analysis Evaluation.....	98
4.5.2 Answer Synthesis Evaluation	104
4.6 Limitations and Future Plans	110
4.7 Conclusion	112
5. BioQA’s Algorithmic Framework.....	113
5.1 Named Entity Recognition	114
5.2 Question Analysis.....	116
5.3 Concepts and Snippets Retrieval.....	118
5.4 Description Generator	120
5.5 Answer Synthesis	121
5.6 Paraphrasing Module	127
5.7 Conclusion	128
6. Concluding Remarks	129
Bibliography	132
Appendices.....	142

Appendix A: Description Templates for Drugbank Entries	142
A.1 Example DrugBank Description Templates	142
A.2 Example DrugBank Generated Descriptions	148
Appendix B: Automated Paraphrasing Rules.....	154
B.1 Simple Substitution Rules	154
B.2 Word Sense Substitution Rules	155
B.3 Enumeration Rules	157
B.4 Rearrangement Rules	157
B.5 Conversion Rules	157
B.6 Other Rules.....	158
Appendix C: Other Information Extraction Techniques in BioQA.....	159
C.1 Chemical Term Recognition	159
C.2 Attribute Extraction.....	160

List of Tables

Table 1: Confusion Matrix showing the evaluation metric for prediction results. TP denotes the number of True-Positives, FP denotes the number of False-Positives and FN denotes the number of False-Negatives.	25
Table 2: Database and Text Collection Statistics for PolySearch2. PolySearch 2.0 significantly expanded the number of text corpora and databases (by >80%) to include a total of 6 free-text corpora and 14 bioinformatics databases. The latest server searches against over 43 million articles covering Medline abstracts, PubMed Central full-text, Wikipedia articles, US Patent abstracts, and open access textbooks.	55
Table 3: PolySearch2 Thesaurus Statistics. PolySearch 2.0 significantly expanded custom thesauri from 9 to 20 categories, and from just 3000 to over 1.13 million term entries. In particular, we have expanded the thesauri to include toxins, food metabolites, biological taxonomies, pathways, as well as Gene Ontology, MeSH terms, and ICD-10 codes. The thesauri also feature many manually curated terms and synonyms for health effects, drug effects, adverse effects, and chemical taxonomies. This table summarizes the number of term entries and synonyms for each thesaurus.	57
Table 4: Performance evaluation of PolySearch2 vs. PolySearch. P stands for Precision, R stands for Recall, F stands for F-measure, and Accu. Stands for accuracy.	61
Table 5: Performance evaluation and feature comparison of PolySearch2 vs. PolySearch.	62
Table 6: Statistics for BioKBs biomedical thesauri collections. This table shows the name of the individual thesaurus, number of terms and synonyms, as well as the primary source. BioKB's thesauri includes terms and synonyms for 20 different types of biomedical entities, including genes, proteins, protein families, diseases, human metabolites, drugs and drug metabolites, biological pathways, tissues, organs, sub-cellular organelles, toxins, food constituents, biological taxonomies, ICD-10 medical codes, positive and adverse health effects, drug effects, and chemical taxonomies.	79

Table 7: Statistics for BioKB’s free-text document collections. This table shows the name of document collections, the number of entries in each document collection, as well as the storage size.	82
Table 8: Statistics for BioKB’s structured database collections. This table shows the name of the database and the number of entries in each database.....	82
Table 9: Statistics for BioKB’s knowledge graph. This table shows the name of each knowledge node, the number of node entries, the number of node attribute fields, the number of internal links (between nodes of same types), and external links (between nodes of different types).	83
Table 10: Example Question Analysis results for the question “What is aspirin?”.....	88
Table 11: Example BioTagger result for the input question “What is aspirin?”.	90
Table 12: Description Generator results for the question “What is aspirin?”.....	91
Table 13: Example of a Concept Graph Generator output on the input question “What is aspirin?”. A subset of 10 edges in the concept graph (51 nodes, 44 edges) are shown in this table. This table shows the concept ID, node type, and node name for source and target nodes for selected edges.....	92
Table 14: Summarization engine output for the question “What is aspirin?”.	93
Table 15: Example Paraphrasing Engine output for synthesized answers with input question “What is aspirin?”.	94
Table 16: Performance statistics of BioQA’s question type prediction algorithm (prefix rule) in comparison with K-nearest neighbor (KNN), Support Vector Machine (SVM), and Random Forest classifiers on the BioASQ training dataset with 600 questions with question type labels.	100
Table 17: BioASQ Challenge Task B Exact answer formation. This table shows the performance statistics for BioQA v1.1 in Task1b, and BioQA v1.2 in Task 2b. Stric Acc. and Lenient Acc. stands for Strict Accuracy, and Lenient Accuracy respectively. MRR stands for mean reciprocal	

rank. Official ranking measures for each answer category are marked with asterisks. Those measures for which BioQA’s overall performance was significantly better than the best among other participants are shown in **bold**..... 106

Table 18: BioASQ Challenge Task B Ideal answer formation. This table shows performance statistics for BioQA v1.1 in Task1b, and BioQA v1.2 in Task 2b. Manual scores for Task 2 were not available. Those measures for which BioQA’s overall performance scores were significantly better than the best among other participants are shown in **bold**. 107

Table 19: Example sentence templates in a group and generated description for a DrugBank entry DB00680 Moricizine. 119

Table 20: Percent (%) coverage for selected data fields in the prokaryotic phenotype database in BacMap and MetaGenAssist. The phenotype database contains a total of 38 data fields (14 shown here) for 10,835 prokaryote species, subspecies and strains. 161

Table 21: An example of potential health effects extracted from MEDLINE abstracts for curcumin, a phytochemical found in the popular Indian spice turmeric. This table lists examples of potential health effect (extracted using the in-house attribute extractor), their scores in co-occurrence analysis, and supporting evidence from reference publications. 162

List of Figures

Figure 1: Number of indexed PubMed (Medline) articles by year.	2
Figure 2: Schema of biocuration workflows and the application of text mining.....	3
Figure 3: Search engine (Google) results for questions “What is the cause of beri-beri?” and “What diseases are caused by E-cadherin mutations?”.	5
Figure 4: Search engine (Bing) results for questions “What is the cause of beri-beri?” and “What diseases are caused by E-cadherin mutations?”.	6
Figure 5: Knowledge engine (Wolfram Alpha) results for questions “What is the cause of beri-beri?” and “What diseases are caused by E-cadherin mutations?”.	7
Figure 6: flow chart diagram showing steps in processing a text collection with various Natural Language Processing techniques.	13
Figure 7: Example of a syntactic parse tree with POS tags, a dependency tree, and semantic role labels for an example sentence “ATP synthase converts ADP to ATP.” Tag abbreviations: S (sentence), NP (noun phrase), VP (verb phrase), PP (prepositional phrase). NNP (singular proper noun), VBZ (verb), IN (preposition).....	17
Figure 8: Illustration of the similarity measure between to document vectors in a vector space model.....	21
Figure 9: General architecture of a QA system. This figure is based on a figure found in “Athenikos, S.J., and Han, H. (2010) Biomedical question answering: A survey. Computer Methods and Programs in Biomedicine, 99(1):1-24, July 2010.”	34
Figure 10: A screenshot of PolySearch2's query interface showing the PolySearch2 query submission form.....	46
Figure 11: A screenshot of PolySearch2's query interface showing the advanced option page for further query refinement.	47

Figure 12: A screenshot of PolySearch2's result display showing the PolySearch2 result overview table.	48
Figure 13: A screenshot of PolySearch2's result display showing the detailed result page with supporting evidence for a single association (Bisphenol A – Breast Neoplasm).	49
Figure 14: A screenshot of PolySearch2's result display showing result details with the full MEDLINE abstract in highlighted and hyperlinked text.	50
Figure 15: PolySearch2's system overview showing the architecture of the PolySearch2 web server, its API, and the underlying search engine. PolySearch2 uses the Model-View-Controller (MVC) design: 1) the PolySearch2 Search Engine with ElasticSearch (Model layer) organizes document collections. 2) the PolySearch2 API (Controller layer) implements the core PolySearch2 algorithms and queries the model layer for search results. 3) the PolySearch2 web server (View layer) is a thin layer of graphical user interface that passes user queries to the PolySearch2 API and formats search results.	53
Figure 16: BioQA’s Query submission page (the Question is: “What is the cause of beri-beri?”).	71
Figure 17: BioQA’s Answer Synopsis page with links to the full answer with references, relevant concepts, the results download, and various knowledge graph visualizations (the Question is: “What is the cause of beri-beri?”).....	72
Figure 18: BioQA’s full answer page (the Question is: “What is the cause of beri-beri?”).....	73
Figure 19: BioQA’s full answer page (the Question is: “What diseases are caused by E-cadherin mutations?”).....	74
Figure 20: BioQA’s relevant concept view for the input question “What diseases are caused by E-cadherin mutations?”.....	75
Figure 21: BioQA’s Co-occurrence network visualization. (The question is: “What diseases are caused by E-cadherin mutations?”)	76

Figure 22: A close-up view on BioQA’s Co-occurrence network visualization. (The question is: “What diseases are caused by E-cadherin mutations?”).....	77
Figure 23: BioQA's knowledge and algorithmic components.....	86
Figure 24: BioASQ Question Similarity Scatter plots: Yes/No questions versus Associative questions.	101
Figure 25: BioASQ Question Similarity Scatter plots: Yes/No questions versus Descriptive questions.	102
Figure 26: BioASQ Question Similarity Scatter plots: Associative question versus Descriptive questions.	103
Figure 27: A flow chart showing BioQA's algorithms and the data flow through the system. ..	113
Figure 28: An example MEDLINE abstract tagged by BioTagger. Surface text tokens recognized as biomedical entities are tagged, color coded, and hyperlinked to corresponding database records.....	115
Figure 29: BioQA's algorithm on summarization via the co-occurrence concept graph.....	122
Figure 30: The Build-Concept-Graph algorithm builds concept graphs from relevant text snippets.	123
Figure 31: BioQA’s summarization algorithm using document matrix and Latent Semantic Indexing techniques.	124
Figure 32: BioQA's automatic summarization algorithm for building a vector space model from retrieved text snippets.	125

List of Abbreviations

CRF	Conditional Random Field
ECMDB	E. coli Metabolome Database
GO	Gene Ontology
HMDB	Human Metabolome Database
HMM	Hidden Markov Model
HPRD	Human Protein Reference Database
IE	Information Extraction
IR	Information Retrieval
IUPAC	International Union of Pure and Applied Chemistry
Jochem	The joint chemical dictionary
KEGG	Kyoto Encyclopedia of Genes and Genomes
LSI	Latent Semantic Index
MeSH	Medical Subject Headings
NCBI	National Center for Biotechnology Information
NE	Named Entity
NER	Named Entity Recognition
NLP	Natural Language Processing
OMIM	Online Mendelian Inheritance in Man
PAS	Predicate Argument Structure
POS	Part-of-Speech
PPI	Protein Protein Interaction
QA	Question Answering
SNOMED-CT	Systematized Nomenclature of Medicine (Clinical Terms)
SRL	Semantic Role Labelling
SVM	Support Vector Machine
UMLS	United Medical Language System
OOV	Out-of-vocabulary

List of My Related Publications

- [1] Liu, Y., Liang, Y., Wishart, D. (2015) PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Research*. Jul 1;43(W1):W535-42.
- [2] Wishart, D., Arndt, D., Pon, A., Sajed, T., Guo, A.C., Djoumbou, Y., Knox, C., Wilson, M., Liang, Y., Grant, J. et al. (2015) T3DB: the toxic exposome database. *Nucleic Acids Research*, 43, D928-934.
- [3] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A.C., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., Neveu, V. et al. (2014) DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, 42, D1091-1097.
- [4] Jewison, T., Su, Y., Disfany, F.M., Liang, Y., Knox, C., Maciejewski, A., Poelzer, J., Huynh, J., Zhou, Y., Arndt, D., Djoumbou, Y., Liu, Y., Deng, L., Guo, A.C., Han, B., Pon, A., Wilson, M., Rafatnia, S., Liu, P., Wishart, D.S. (2014) SMPDB 2.0: big improvements to the Small Molecule Pathway Database. *Nucleic Acids Research*. Jan;42(Database issue):D478-84.
- [5] Wishart, D.S., Jewison, T., Guo, A.C., Wilson, M., Knox, C., Liu, Y., Djoumbou, Y., Mandal, R., Aziat, F., Dong, E. et al. (2013) HMDB 3.0--The Human Metabolome Database in 2013. *Nucleic Acids Research*, 41, D801-807.
- [6] Guo, A.C., Jewison, T., Wilson, M., Liu, Y., Knox, C., Djoumbou, Y., Lo, P., Mandal, R., Krishnamurthy, R. and Wishart, D.S. (2013) ECMDB: the E. coli Metabolome Database. *Nucleic Acids Research*, 41, D625-630.
- [7] Arndt, D., Xia, J., Liu, Y., Zhou, Y., Guo, A.C., Cruz, J.A., Sinelnikov, I., Budwill, K., Nesbo, C.L., Wishart, D.S. (2012) METAGENassist: a comprehensive web server for comparative metagenomics. *Nucleic Acids Research*. Jul;40(Web Server issue):W88-95.
- [8] Cruz, J., Liu, Y., Liang, Y., Zhou, Y., Wilson, M., Dennis, J.J., Stothard, P., Van Domselaar G., Wishart, D.S. (2012) BacMap: an up-to-date electronic atlas of annotated bacterial genomes. *Nucleic Acids Research*. Jan;40(Database issue):D599-604.

1. Introduction

This chapter introduces the concept of biomedical question answering, presents the thesis statement and outlines the rest of this thesis regarding my approach to accomplish the task of biomedical question answering.

1.1 Introduction to biomedical question answering

Biomedical researchers are now finding it almost impossible to stay current with today's research due to the vast amount of data being generated and published. Consider these facts: 1) there are more than 60,000 scientific journals published today; 2) nearly 500,000 biomedical articles are published each year; 3) there are more than 22,000,000 abstracts in PubMed published from 20,000 journals; and 4) there have been more than 50,000,000 scientific articles published since 1660. According to a study by Baasiri *et al.* [9] a researcher would have to scan 130 different journals and read 27 papers per day to follow a single disease, such as breast cancer. A more recent study by Lu *et al.* [54], showed that the total number of citations in MEDLINE, a central repository for scientific articles in the biomedical domain, is growing at more than 4% each year, and that more than 3,000 new articles are being added each day. Figure 1 shows the number of indexed articles in MEDLINE and the accelerating growth rate of the PubMed database [19]. In addition to the rapid growth in published biomedical literature, biomedical databases are growing too. GenBank [18], which contains most of the world's gene sequencing information, has grown from just 600 annotated DNA sequences in 1982 to nearly 200 million annotated DNA sequences today. The Protein Data Bank [69], which houses most of the world's protein structure data, grew from 13 structures in 1976 to more than 120,000 structures by 2015. ArrayExpress [70], which contains data on gene expression experiments, grew from just 1,200 data sets in 2006 to nearly 70,000 today. Adding to the challenge of exponential information growth, is the proliferation of domain-specific databases. For instance, the total number of biomolecular databases ever described in the annual Nucleic Acids Research (NAR) Database Issue has grown from 90 in 1998 to nearly 1,700 today [80]. These data show that it is increasingly difficult for biomedical researchers to keep up with current research, let alone learn from past research results. It is also evident that a considerable amount of useful biological knowledge is buried in the form of free text, waiting to be transformed into more

accessible formats. Swanson referred to such phenomena as “undiscovered public knowledge” [13].

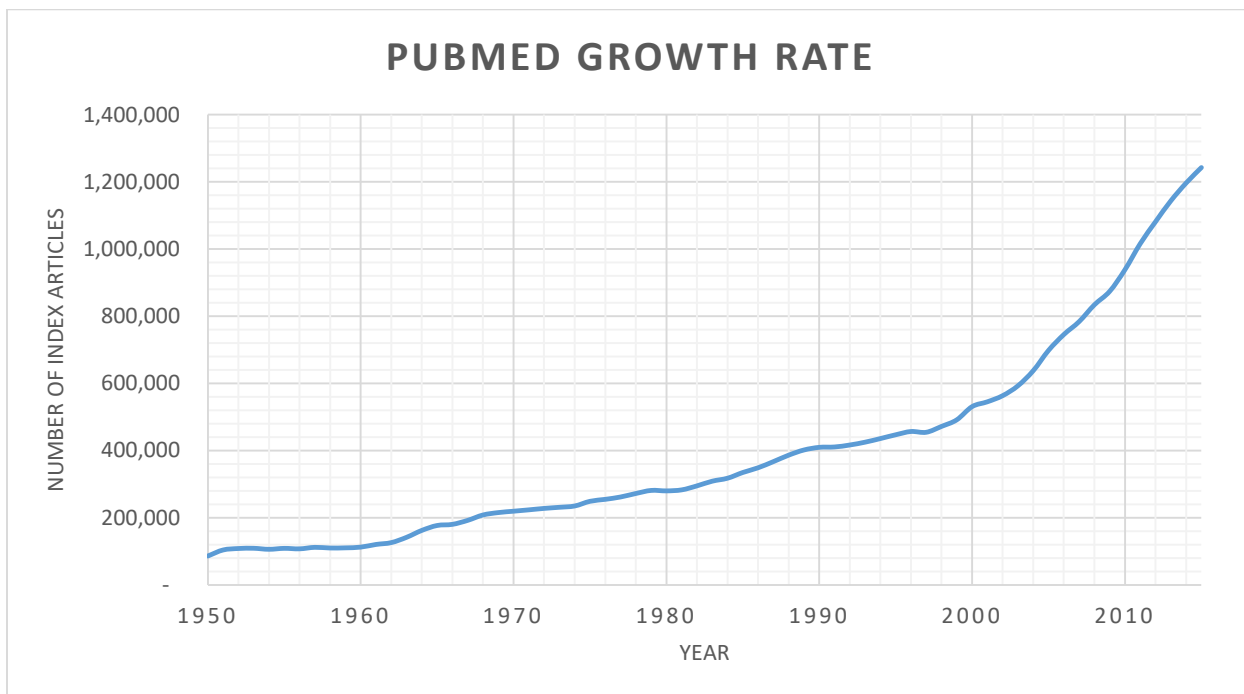


Figure 1: Number of indexed PubMed (Medline) articles by year.

To alleviate the problem of information overload, teams of biocurators are increasingly being employed to convert paper-bound text into electronically accessible information through the construction of biomedical databases. These databases are frequently serving as the backbone of the field’s working knowledge. Biocuration [40] aims at organizing and annotating discoveries disseminated by biological researchers. An important aspect of biocuration is that useful knowledge in published (i.e. paper) articles is assembled and deposited into electronic biomedical knowledge bases that are accessible over the internet. However, biocuration is a time-consuming and labor-intensive process that requires the effort of many high-priced domain experts.

Computer-aided text mining can serve as an important means of reducing the biocuration bottleneck as it enables biomedical researchers to rapidly and automatically retrieve existing knowledge or discover hidden knowledge in the literature. To date, text mining approaches in biomedicine have focused on such tasks as: 1) retrieving relevant documents, 2) extracting mentioned biomedical entries and predicting their association (e.g. Protein-Protein Interactions), 3) learning or enriching biomedical ontologies from text, and most recently, 4) providing salient answers for clinical questions [7].

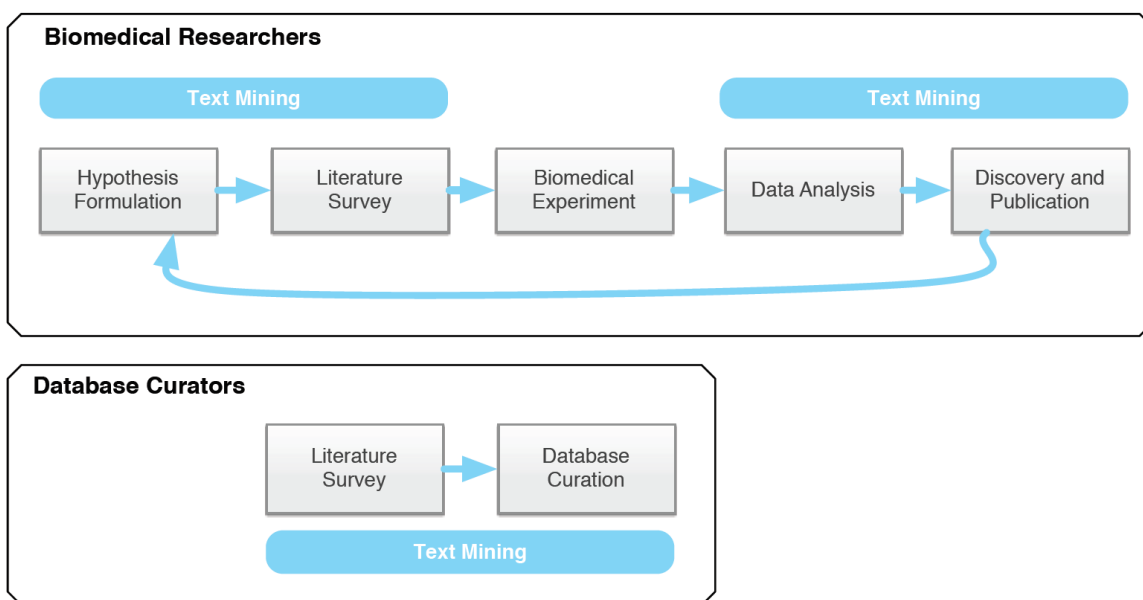


Figure 2: Schema of biocuration workflows and the application of text mining.

Figure 2 illustrates an example of a typical biomedical discovery workflow. In this workflow a biomedical scientist first formulates a hypothesis, then they conduct a literature search to find any prior (published) knowledge that may help test the hypothesis or enhance their understanding on the subject matter. They then conduct their own experiments to test the hypothesis and report their discoveries in a peer reviewed scientific publication. In addition to the traditional route of journal publication, a growing number of scientific journals require that researchers deposit their experimental data into publicly accessible databases prior to

publication. However, snippets of scientific discoveries (such as concentrations of a newly discovered metabolite in body fluids) are often buried in the free text of the published article and not in an easily accessible abstract. Such knowledge could be extracted manually or semi-automatically by biocurators and finally deposited into dedicated databases. Text mining facilitates scientific discovery and biocuration by first providing a means of retrieving relevant articles for literature review and hypothesis formulation, then suggesting hidden associations that may have been previously overlooked by researchers (or biocurators). To illustrate the potential of text mining in biomedical research, Swanson [13], a mathematician, conducted keyword searches on MEDLINE and examined shared terms to formulate hypotheses for previously overlooked relations between seemingly disconnected topics, such as magnesium and migraine, fish oil and Raynaud's syndrome, and somatomedin C and arginine. These novel, text-mined relationships were later supported by biological experiments or clinical trials [13].

1.2 Challenges


Recent attention towards biomedical text mining has focused on developing improved search engines and providing easier ways to navigate biomedical publications. These engines use ontologies to recognize biomedical named entities (NEs) from text, extract explicit relations between entities, construct domain specific lexicons to support other text mining tasks, and curate datasets for evaluating text mining techniques. Despite continuing advances and improvements in biomedical text mining, a publicly accessible, domain-specific question answering (QA) system is still not available. The central idea behind a biomedical QA system is to offer structured, precise and salient answers to natural language biomedical questions posed by researchers and biocurators. Such a QA system would benefit biomedical researchers, physicians, students, and the inquisitive public. Over the past decade we have witnessed huge advances in text mining applications including the rise of search engines like Google [68], Bing [60], and knowledge engines like Wolfram Alpha [99]. In February 2011, the IBM-developed QA system called Watson [28] defeated highly skilled human players on the open-domain question answering *Jeopardy!* challenge.

Google

All Images Videos Shopping News More Search tools

About 487,000 results (0.84 seconds)

Beriberi is a disease caused by a vitamin B1 (thiamine) **deficiency**. There are two types of the disease: wet beriberi and dry beriberi. Wet beriberi affects the heart and circulatory system. In extreme cases, wet beriberi can cause **heart failure**. Nov 23, 2015



[Beriberi: Overview, Causes & Symptoms - Healthline](http://www.healthline.com/health/beriberi)
www.healthline.com/health/beriberi Healthline Networks

[About this result](#) • [Feedback](#)

[Beriberi: Overview, Causes & Symptoms - Healthline](http://www.healthline.com/health/beriberi)
www.healthline.com/health/beriberi Healthline Networks

Nov 23, 2015 - Beriberi is a disease caused by a vitamin B1 (thiamine) deficiency. There are two types of the disease: wet beriberi and dry beriberi. Wet beriberi affects the heart and circulatory system. In extreme cases, wet beriberi can cause heart failure.
[Overview](#) · [Causes and Risk Factors](#) · [Symptoms](#) · [Diagnosis](#)

Google

All Images News Videos Shopping More Search tools

About 1,380,000 results (0.54 seconds)

[Cadherins as Targets for Genetic Diseases](#)
www.ncbi.nlm.nih.gov/.../PMC28278... National Center for Biotechnology Information
by A El-Amraoui - 2010 - Cited by 39 - [Related articles](#)
Mutations in P-cadherin were later reported to cause another autosomal recessive ... These cadherins resemble the classical cadherins, such as E-cadherin.

[CDH1 gene - Genetics Home Reference](#)
https://ghr.nlm.nih.gov/gene/CDH1 United States National Library of Medicine
3 days ago - The CDH1 gene provides instructions for making a protein called epithelial cadherin or E-cadherin. This protein is found within the membrane ...

[E-Cadherin and Gastric Cancer: Cause, Consequence, and Applications](#)
https://www.hindawi.com/journals/bmri/2014/637308/ by X Liu - 2014 - Cited by 27 - [Related articles](#)
Jul 31, 2014 - Genetic Mutations and Variants of E-Cadherin in Diffuse Gastric Cancer ... It has been reported that H. pylori infection is associated with CDH1 ...

Figure 3: Search engine (Google) results for questions “What is the cause of beri-beri?” and “What diseases are caused by E-cadherin mutations?”.

what is the cause of beri-beri

168,000 RESULTS Any time

Causes of Thiamine Deficiency:

- Cancer Patients treating with DCA
- Crash Dieting
- Liver Disfunction
- Kidney Dialysis
- Alcohol Abuse
- Consumption of sweets, soft drinks, processed foods
- Sustained periods of IV nutrients
- Carbohydrate Heavy Diet

[Thiamine Deficiency: List of Symptoms, Causes, Reversing ...](#)
thiaminedeficiency.org/

Is this answer helpful?

Beriberi - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Beriberi
Cause. Beriberi may also be caused by shortcomings other than inadequate intake: ... In the first known description of beriberi (or, beri-beri), ...

Beriberi - Symptoms, Causes, Treatments - Healthgrades
www.healthgrades.com/conditions/beriberi
Beriberi Information Including Symptoms, Diagnosis, Treatment, Causes, Videos, Forums, and Local Community Content. Find answers to health issues you can trust.

Beriberi
Beriberi refers to a cluster of symptoms caused primarily by thiamine deficiency. Beriberi has conventionally been divided into three separate entities, relating to the body system mainly involved or age of person. Beriberi is one of several thiamine-deficiency related conditions, which may occur concurrently, including Wernicke's encephalopathy, Ko...

Wikipedia

People also search for

- Pellagra
- Scunvy
- Rickets
- Wernicke-Korsakoff syndrome
- Kwashiorkor

See more

Data from: Wikipedia
Feedback

Beriberi Disease Symptoms | Lifescript.com
Ad - Lifescript.com/Health
31,300+ followers on Twitter
Find Facts, Symptoms & Treatments. Trusted By 50 Million Visitors.

What diseases are caused by e-cadherin mutations?

4,300,000 RESULTS Any time


CDH1 gene - Genetics Home Reference
ghr.nlm.nih.gov/Genes
... without a family history of the disease. ... HDGC caused by CDH1 gene mutations are born with one ... F, Galimberti V. E-cadherin germline mutation ...

E-cadherin germline mutations in familial gastric ...
www.ncbi.nlm.nih.gov/pubmed/9537325
E-cadherin germline mutations in familial gastric cancer. ... and confirm the important role of E-cadherin mutations in ... to Disease; Germ-Line Mutation" ...
Published in: Nature - 1998
Authors: Parry Guilford · Justin Hopkins · J R Harraway · Maybelle Mcleod · Nga...
Affiliation: University of Otago
About: Ecology · Developmental biology · Evolution · Cancer · Neuroscience · ...

Germline E-cadherin Gene (CDH1) Mutations Predispose ...
hmg.oxfordjournals.org/content/8/4/607.full
Apr 08, 2015 · Germline E-cadherin ... in tumour sections from families where an E-cadherin mutation was not detected and is consistent in ... diseases; Genetics of ...

E- cadherin mutation - uptodate.com
www.uptodate.com/contents/search?search=E.+cadherin+mutation&x=0&y=0
... Gastroenterology and Hepatology, Hematology, Infectious Diseases, Nephrology ...
Search Results for "E- cadherin mutation" ... Eosinophil biology and causes of ...

Figure 4: Search engine (Bing) results for questions “What is the cause of beri-beri?” and “What diseases are caused by E-cadherin mutations?”.

 computational knowledge engine

What is the cause of beri-beri | ☆

Web Apps Examples Random

! Wolfram|Alpha doesn't understand your query ?


Showing instead result for query: **beri beri**

Assuming Chewong | Use Zaghawa instead

Input interpretation:
Chewong (language) | **Chewong** (language)

Basic properties:

	Chewong	Chewong
total number of speakers	200 people <small>(2005 estimate)</small>	200 people <small>(2005 estimate)</small>

 computational knowledge engine

What diseases are caused by e-cadherin mutations? | ☆

Web Apps Examples Random

! Wolfram|Alpha doesn't understand your query ?

Showing instead result for query: **e-cadherin**

mutations

Assuming Cdh1 (mouse gene) | Use Cad99C (fruit fly gene) or more instead

Input interpretation:
Cdh1 (mouse gene)

Standard name:
cadherin 1

Figure 5: Knowledge engine (Wolfram Alpha) results for questions “What is the cause of beri-beri?” and “What diseases are caused by E-cadherin mutations?”.

The success of Watson has motivated text mining researchers to start thinking about developing question-answering systems for biomedicine [100]. However, biomedical QA is challenging as it is a highly specialized domain. It is somewhat different from open domain QA, and the supporting evidence is often stored in heterogeneous sources in various formats that cannot yet be searched like web pages on a traditional search engine without explicit consolidation and curation. To illustrate the challenge in Biomedical QA, we posed the following questions “*What is the cause of beri-beri?*” and “*What diseases are caused by E-cadherin mutations?*” to Google [68], Bing [60], and Wolfram Alpha [99]. The results are shown in Figure 3, Figure 4, and Figure 5. With the first question “*What is the cause of beri-beri?*”, Google [68] is able to return a text snippet (extracted from the top hit) identifying beri-beri as a disease caused by thiamine deficiency. Bing [60] is able to return the cause as being thiamine deficiency and lists different causes for thiamine deficiency. Wolfram Alpha [99], a knowledge engine, is not able to identify beri-beri as a disease but instead assumes it is a language and presents regions and number of speakers for the assumed language. With the second question “*What diseases are caused by E-cadherin mutations?*”, both Google and Bing interpret the question as a search query and return web pages containing information on E-cadherin and the CDH1 gene. The third hit from Google and second hit from Bing indicate a connection between E-cadherin and Gastric Cancer but no further information is provided for the potential association. Top hits from both search engines now include biomedical publications, but no answer snippets or other pieces of evidence are extracted from these publications. Wolfram Alpha [99] is capable of mapping E-cadherin to a mouse gene (CDH1) and provides the name and genetic sequence for this gene. However, no disease with potential associations to E-cadherin are shown in Wolfram Alpha’s results. For this question, none of these systems are capable of providing descriptions for the concept of “E-cadherin”, and none of them is able to provide natural language answers to this question. This little QA experiment illustrates the need to develop a specialized biomedical question answering system, capable of accepting questions in natural language sentences, and providing natural language answers for the individual posing the question.

1.3 Thesis Statement

Over the past two decades a number of keyword-query document retrieval systems (i.e. PubMed) have been developed to help alleviate the problems associated with biomedical question-answering. However, users still have to manually examine the retrieved documents to find answers to their queries. Ideally what is needed is a “wise” question answering (QA) system that uses natural language and figures out what the questioner is really asking and composes natural language answers.

My thesis research focuses on developing a biomedical question answering system (called BioQA) that would provide precise, natural language answers with encyclopedia-like commentary to a wide range of natural language biomedical questions. In particular, this biomedical question answering system should be able to handle both descriptive (“*What is Aspirin?*”) and associative queries (“*What is the cause of beri-beri?*”). Descriptive queries are particularly useful for automatically creating annotations of genes/proteins for newly sequenced organisms while associative queries are useful for discovering relations between biomedical entities. This QA system should also be able to summarize relevant documents and text passages using natural language and generate supporting evidence for the returned answers. The design of BioQA focuses on answering biomedical questions posed by researchers, physicians, students and the inquisitive public.

In this thesis, I hypothesize that building a biomedical question answering system is feasible and I propose the BioQA framework with a prototype implementation to demonstrate the feasibility and usability of a QA system in biomedicine. Through the implementation of BioQA, I learned that 1) a comprehensive biomedical thesaurus is essential for almost all steps of biomedical question answering, and 2) effective summarization algorithms are the key to derive natural language answers from relevant concepts and snippets.

1.4 Thesis Outline

In this thesis, I present a framework for building a practical biomedical question answering system. To illustrate the feasibility of such framework, I developed a prototype QA system, called BioQA. To demonstrate its usability in answering biomedical questions and to serve the general public I also created a publicly accessible web interface for BioQA and one of its search engines (PolySearch2). The rest of this dissertation is organized as follows:

Chapter 2 provides an overview on text mining techniques relating to question answering. This chapter serves as a brief review and introduces related concepts in natural language processing, machine learning, information retrieval and information extraction, as well as various evaluation metrics. This chapter also discusses related works in biomedical question answering, including biomedical thesauri and ontology curation. It also describes recent developments in document retrieval, named entity recognition, ontology matching, relation extraction, and automated question answering.

Chapter 3 presents PolySearch2 [52], a biomedical association extraction system and biomedical domain-specific search engine. PolySearch2 is designed to identify latent relationships between biomedical entities as well as mining reference snippets as evidence for discovered associations. This chapter also introduces PolySearch2's public web interface, its enhancements over the legacy PolySearch [16, 17] system, its underlying methodologies and its performance evaluation. PolySearch2 served as a precursor to the development of BioQA.

Chapter 4 presents BioQA, an automated biomedical question answering system. In this chapter I propose a biomedical question-answering framework (the BioQA framework) and describe how BioQA was assembled and tested. In this chapter, I describe BioQA's public web interface, its underlying knowledgebase BioKB, the collection of algorithms for transforming input questions with retrieved knowledge to natural language answer summaries, and discuss its performance evaluations results.

Chapter 5 provides further details on the BioQA's algorithmic framework for named entity recognition, question analysis, concept and snippet retrieval, description generation, answer synthesis, and automated paraphrasing.

Chapter 6 concludes this thesis by reviewing the research contributions of PolySearch2 and BioQA, as well as the future directions for further research in biomedical question answering.

This thesis is accompanied with three appendices. In these appendices, automated description generation (Appendix A), automated paraphrasing (Appendix B), and other information extraction techniques used in the BioQA question answering framework (Appendix C) are described in detail.

2. Background and Related Works

In this chapter, I briefly review a number of text-mining techniques related to question answering (QA). I also introduce related concepts in natural language processing, machine learning, information retrieval and extraction, as well as various evaluation metrics. I also survey a number of related works associated with question answering, specific to the biomedical domain. This includes biomedical thesauri, ontologies, biomedical information retrieval and extraction, and prior studies and reports on automated question answering.

2.1 Text Mining Overview

Text mining utilizes various techniques in natural language processing (NLP), supervised machine learning, unsupervised data mining, information retrieval (IR), and information extraction (IE) to extract useful information from free text and format it into a well-defined data structure. This section provides a brief overview of the key methodologies and applications used in text mining.

2.1.1 Natural Language Processing

Text mining can be viewed as a specific application of natural language processing (NLP) techniques [46, 56], which together with other machine learning and data mining methods, can be used to discover useful information hidden in raw text. NLP provides the basic tools for analyzing the semantics (or meaning) of a sentence. Processing results from various NLP algorithms provide additional information for downstream, supervised machine learning or unsupervised data mining. Text mining uses NLP techniques in almost all levels, including but not limited to, language modelling, Part-of-Speech (POS) tagging, syntactic parsing, semantic role labelling, and summarization. Figure 6 shows some of the steps typically used to process documents in preparation for further processing with various natural language processing algorithms. These terms and processes are explained in more detail below.

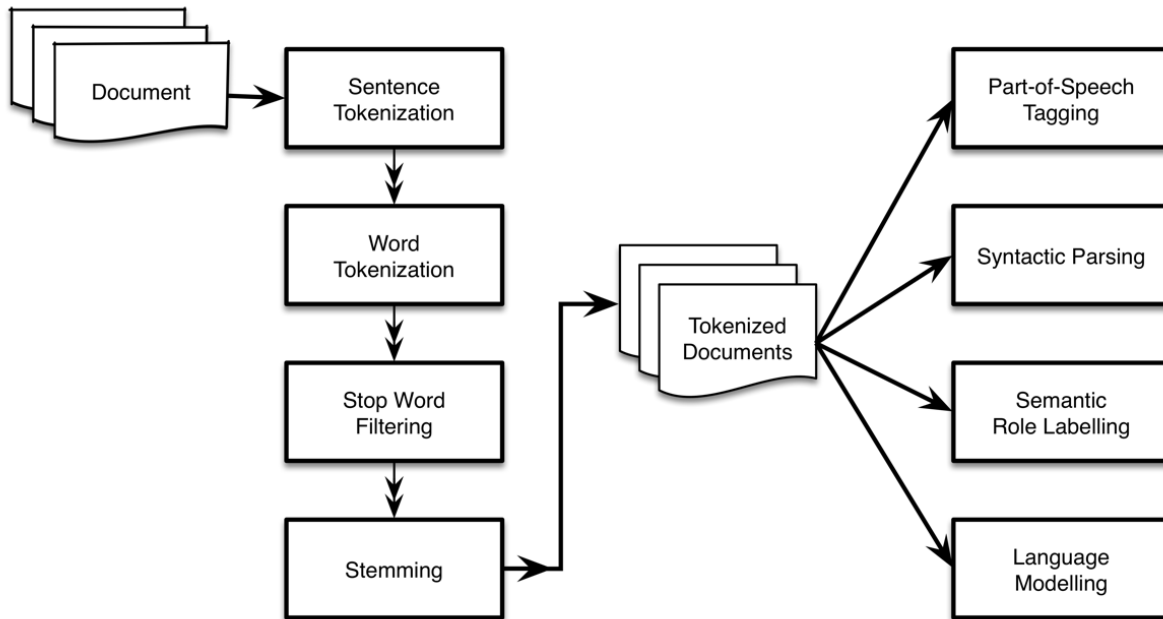


Figure 6: flow chart diagram showing steps in processing a text collection with various Natural Language Processing techniques.

As can be seen in Figure 6, language modeling plays an important role in NLP. A *language model* is a probability distribution of terms occurring across all documents in a text collection. Each text document consists of one or more tokens; each token is a delimited sequence of characters. For example, in the sentence “virus contains DNA.”, we have three tokens “virus”, “contains”, “DNA”, each of which is delimited by white space. Language modelling calculates N-gram (N consecutive tokens) frequencies or skip N-gram (N subsequence of tokens that need not be consecutive) frequencies from a given body of text (or sequence of terms). N-gram language models can be used to provide probability estimates for multiple word terms (N-gram) in a given sentence, and assess its relative importance in a text collection. N-gram frequency calculation is done by counting the occurrence of each unique N-gram in a tokenized text collection, and dividing that count by the total number of tokens in that text collection. For example, in a text collection with two sentences: [“virus contains DNA”, “plants also contain DNA.”], we have six unique tokens [“virus”, “contains”, “contain”, “DNA”, “plants”, “also”]. Each token occurs exactly once, but “DNA” occurs twice, so the probability

that the token “DNA” occurs in the text collection is estimated to be $2/7$, or 0.2857. Notice that tokens “contain” and “contains” are same word in different forms, but are counted as different tokens. We can convert different forms of the same word to a unique base form using *stemming*.

Stemming is a word transformation technique that trims off the suffix of a word so it is reverted back to its common base form (stem). For example, the tokens “trim”, “trims”, “trimming”, and “trimmed” are converted to the same base form “trim” via word stemming. Quite often we calculate N-gram frequency on a stemmed collection of tokens to take into account the fact that words can occur in various forms. For example, after stemming, our example text collection now contains only five tokens [“virus”, “contain”, “DNA”, “plant”, “also”], and both “DNA” and “contain” now occur twice.

In building language models, we also need to filter out stop words from a sentence. Stop words are words that commonly occur in almost every sentence with little significance in probability estimation or language modelling. Some examples of stop words are “is”, “are”, “also”, “will”, “does”, “do”, “as”, “were”, “has”. In our example, after removing the stop word “also” we have a list of four unique tokens [“virus”, “contain”, “DNA”, “plant”], with “DNA” occurring twice and “virus” occurring once. The occurrence probability for “DNA” and “virus”, adjusted after removing the stop words, is therefore $2/6$ or 0.3333, and $1/6$ or 0.1667, respectively. Language modelling also estimates the occurrence frequency for terms that are missing from the corpus (“out-of-vocabulary” or OOV terms) using various “smoothing” techniques. For example, we can assume a word that does not occur in the base text collection will likely occur with a fixed low probability. For example, the word “animal” does not occur in our example text collection and therefore by smoothing we assign it a low occurrence probability of, say, 0.00001.

We can calculate the occurrence probability of a sentence by calculating the cumulative probability of each token in the sentence. Smoothing is needed to ensure the cumulative probability does not reduce to zero due to an OOV word occurring in the sentence. Language modelling techniques are used widely in NLP applications like document retrieval and text clustering. For example, Google uses N-gram language models in its web page retrieval

algorithms. The Google N-gram data [33] provides frequency counts for more than 1.1 billion 5-gram (five consecutive word sequences) calculated from indexed web pages.

Part-of-Speech (POS) tagging assigns POS tags (e.g. nouns, verbs, adjectives, adverbs etc.) to each word in a given sentence. The Penn Treebank Project [58] provides a standardized collection of POS tags that are widely used in NLP. There are both rule-based POS tagging and probabilistic POS tagging approaches. Rule-based approaches attempt to assign a POS tag to a word based on its dictionary entry or via context words in a given sentence. Probabilistic POS tagging trains a probabilistic machine learning model to predict a word's most probable POS tags using probability estimations from a labelled training dataset (prior probability) and the observed sequence of words in the input. For example, the Viterbi algorithm [30] implements a Hidden Markov Model (HMM) to perform POS tagging for a sentence by predicting the most probable sequence of hidden states (POS tags) from a given sequence of observations (words in a given sentence).

Syntactic parsing converts a given sentence into a syntactic parsed tree, which identifies syntactic constituents like noun phrases, verb phrases and prepositional phrases. Semantic role labelling (SRL) identifies arguments of predicates (or verbs) in a given sentence. For example, the semantic roles in the sentence “ATP synthase converts ADP to ATP.” are:

[ATP synthase (A0/Subject)] [converts (Verb)] [ADP (A1/Object)] to
[ATP (A2/Indirect Object)].

Figure 7 shows an example of a POS-tagged sentence that includes a syntactically parsed tree, the dependency relationships between words, as well as the semantic roles for each constituent. As illustrated in Figure 7, this sentence about ATP synthase (the root) consists of a noun phrase (NP) and a verb phrase (VP). The NP consists of two proper nouns “ATP” and “synthase”, and the VP consists of the main verb “converts” and another NP. The NP consists of a proper noun “ADP” and a prepositional phrase (PP) “to ATP”. The syntactically parsed tree can therefore be converted into a dependency tree illustrating the dependencies between words. In this case, the sentence root depends on the main verb “converts”, and the verb is dependent on both the subject “ATP synthase” and the objects (“ADP” and “to ATP”) of the sentence.

Semantic role labelling is an important step needed to understand the semantics or meaning of a sentence. This process provides a higher level of abstraction than a simple syntax tree. This is because semantic role labeling can be used to convert sentences with the same semantics but different syntactic variations into the same canonical Predicate Argument Structure (PAS) [56]. For example, the following sentence can be written in many forms with different syntactic variations: "ADP is converted to ATP by ATP synthase.", "ATP is converted from ADP by ATP synthase.", "By ATP synthase, ADP is converted to ATP". However, the underlying canonical PAS for all three sentence variants "convert([ADP], [ATP], [ATP synthase])" would be identical.

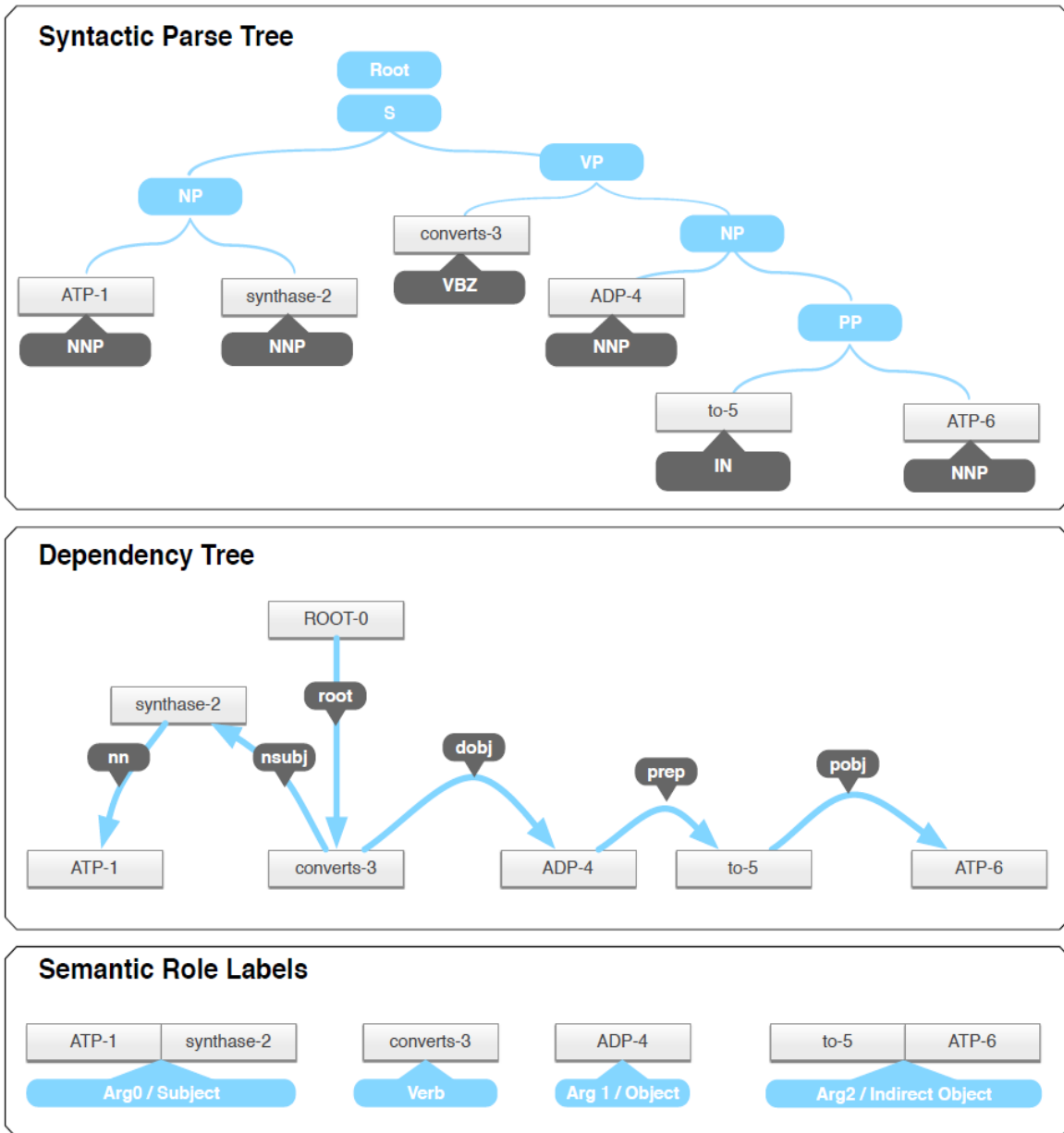


Figure 7: Example of a syntactic parse tree with POS tags, a dependency tree, and semantic role labels for an example sentence “ATP synthase converts ADP to ATP.” Tag abbreviations: S (sentence), NP (noun phrase), VP (verb phrase), PP (prepositional phrase). NNP (singular proper noun), VBZ (verb), IN (preposition).

2.1.2 Machine Learning

Machine Learning (ML) is a subfield of computing science that focuses on building mathematical models from example training data and uses those models to assign new input data instances (supervised machine learning) or explore characteristics in data without class labels (unsupervised machine learning). Supervised machine learning is capable of empirically learning classification or regression models from labelled training data (a training set) using statistical methods. Once properly trained, machine learning models can predict class labels (classification) or values (regression) on novel or unseen data (testing set) based on the learning models. Both classification and regression tasks are supervised, as they require the input of labelled training data. Classifications, in particular, are commonly used in many information retrieval, information extraction, and natural language processing applications. These include part-of-speech tagging, named entity recognition, sentence chunking, syntactic parsing and semantic role labelling. Some of the more popular supervised machine learning approaches used in text mining include Support Vector Machines (SVMs), Hidden Markov Models (HMMs), Naive Bayes Classifiers, and Conditional Random Fields (CRF) [56].

While classification and regression require labelled training data, unsupervised machine learning or data mining [85] does not require such labelled training data. Data mining techniques, such as cluster analysis and association analysis, are capable of discovering useful patterns from unlabeled data. Cluster analysis assigns data objects to groups called clusters, based on data object attributes and a defined measure of their similarity or difference. Popular clustering methods include K-means clustering, agglomerative hierarchical clustering, and density-based clustering [85]. For example, clustering analysis can be used to organize sentences in a paragraph into groups based on the terms in each sentence. Association analysis discovers interesting relationships hidden in large data sets. The discovered relationships take the form of association rules, which map a set of data objects to another associated data object with a certain degree of support and confidence. An advantage of data mining over supervised machine learning is that data mining is capable of making data-driven inferences without requiring labelled training data. Data mining has many applications in text mining, including document categorization, term mapping to concepts in a target ontology, and discovering implicit connections between concepts.

2.1.3 Information Retrieval

Information retrieval (IR) fetches relevant documents from a document collection in response to user queries expressed in the form of search keywords. Google, a widely used web search engine, can be considered an IR system that retrieves relevant web pages (documents) from all indexed web pages on the World Wide Web (a very large document collection). PubMed, a biomedical article search engine, is yet another example of an IR system that retrieves relevant articles among published biomedical articles using Boolean keyword queries [66, 67]. Document indices are used in IR systems to ensure rapid and effective retrieval.

IR systems typically organize documents in a document collection using an inverted index. An inverted index is a lookup table mapping a keyword in a document collection to the list of documents containing that keyword. IR systems select relevant documents based on user-defined search keywords by fetching documents with index keywords that either match the search keywords or which achieve a certain heuristic matching/relevancy score. IR systems score and rank the retrieved document list according to predefined criteria, and return to the user a ranked list of relevant documents (hits). The criteria for ranking retrieved documents are application specific and can vary from one application to another. For example, Google ranks web pages found using search keywords through link structures in the web using its PageRank [8] algorithm. On the other hand, Google Scholar ranks retrieved publications by citation frequency, and PubMed ranks MEDLINE entries by publication date. IR systems often transform and represent documents in a Vector Space Model [46] for more efficient and accurate document retrieval. A document can be represented as a “bag of words” with certain word frequency counts. This representation transforms a free-text document to a numerical vector, with each element in the vector corresponding to occurrence frequency for a word (or N-Gram) in a given document. The collection of unique words (or N-Grams) occurring in all documents defines the vocabulary. In this model, each word represents a dimension in the Vector Space Model, while the size of the vocabulary dictates length of document vector. This is because each document vector must contain the same number of elements as the vocabulary. A Vector Space Model can have very high dimensionality, quite often on the order of tens or hundreds of thousands of dimensions. This is because each dimension implicitly represents a topic (a key phrase, or word sequence), while each document represents a vector in this space spanning by the topics

(dimensions) it discusses. We use the phrase term frequency (TF) as an occurrence frequency measure in a document vector. Weighting terms based solely on term frequency tends to emphasize highly popular terms and understate rare terms in a document. However, documents can be better distinguished on rare terms that characterize each document. So we also need to consider Inverse Document Frequency (IDF), which measures the rareness and importance of a term according to the number of documents containing such a term. The intuition is that a term occurring in many documents is less discriminative than another term occurring in only a few documents. Term Frequency - Inverse Document Frequency (TF-IDF) weighting balances term weighting by combining both Term Frequency measure and Inverse Document Frequency,

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

In the above formula, $\text{tf}_{t,d}$ is the occurrence frequency of term t in document d , and idf_t is inverse document frequency of term t , which is defined as:

$$\text{idf}_t = \log \frac{N}{|\{d \in D : t \in d\}|}$$

where N is the number of documents in the collection, and $|\{d \in D : t \in d\}|$ is the number of documents d containing the term t .

Representing documents through a Vector Space Model enables us to perform vector calculations for documents. For example, we can calculate the similarity between two documents using the Cosine similarity measure. Since each document is a vector in the Vector Space Model, we can define a similarity measure between two documents according to the Cosine value of the angle formed by their corresponding document vectors:

$$\text{similarity}(A, B) = \cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

In the above formula, A and B are document vectors, and θ is the angle between A and B . $\cos(\theta)$ represents the similarity between documents A and B . A_i and B_i are occurrence frequency measures (e.g. TF-IDF) for word i (in the vocabulary of size n) in vector A and B , respectively. Figure 8 shows a conceptual illustration for the angles between two document

vectors in a vector space models. The angle θ between the two documents defines their similarity. Cosine similarity is bounded by the interval $[0, 1]$. Therefore $\cos(\theta) = 1$ when two document vectors are identical or very highly similar, and $\cos(\theta) = 0$ when two document vectors are completely opposite or maximally dissimilar to each other.

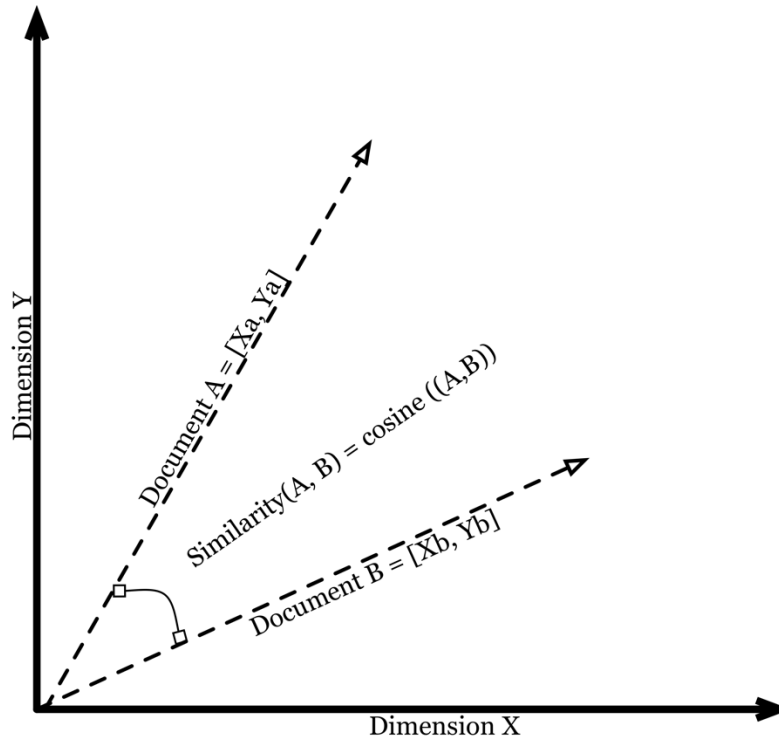


Figure 8: Illustration of the similarity measure between to document vectors in a vector space model.

Information retrieval serves as a basic building block for many text mining systems, so there is a clear need for an effective IR system in any practical text mining project. Recently, the Apache Software Foundation provided developers with an open source IR engine through the Apache Lucene project [38]. Lucene utilizes an efficient data structure (the Lucene index) to index and organize gigabytes of text documents onto a hard disk. Elasticsearch [71], an enterprise search engine built on top of Lucene, improves scalability and reliability of document retrieval by using a collection of distributed shards (Lucene indices) with the ability to

dynamically duplicate and shuffle documents between shards for higher reliability and efficiency. Each shard (Lucene index) is a structured collection of documents (JSON objects) formatted and indexed for fast retrieval from disk. An Elasticsearch server is essentially a collection of machines (nodes), each maintaining one or more shards. Elasticsearch dynamically duplicate documents between shards to improve reliability, and it also dynamically shuffles documents to recently requested shards to improve document retrieval efficiency. Lucene and Elasticsearch [71] are just two among many open source IR systems that are making document retrieval much easier and more scalable for open source projects that require efficient retrieval of documents from enormous flat file document collections.

2.1.4 Information Extraction

Information extraction focuses on extracting Named Entities (NEs) and their relationships from surface text. Surface text, which is also known as raw text, are expressions that are actually used in a sentence, and are implicitly used in mapping concepts in a knowledge domain. The task of Named Entity Recognition is the explicit mapping between expressions (surface text) and their semantic meanings (concepts) they represent. In the open domain, information extraction focuses on the extraction of NEs such as names of companies, people, and places. A naive method for named entity recognition (NER) is to use simple dictionary matching. However, dictionary-based term matching is often insufficient to extract all NEs, as one typically does not have the complete knowledge of all entity names, their synonyms, and their various surface forms in written text. Therefore, linguistic or statistical methods, such as rule-based or machine learning-based methods, are used for more precise information extraction. Rule-based systems require the curation of extraction rules, which are derived by domain experts. Alternately, these rules can be generated automatically through machine learning methods followed by manual curation. Machine learning-based approaches require the curation of labelled training data and the choice of classification features and classification models. Both approaches benefit from open domain resources (e.g. WordNet [62]) or domain specific lexicons. Rule-based information extraction systems are usually high in precision but low in recall, while machine learning-based systems are usually high in recall but low in precision. Due to their complementary strengths and

weaknesses, rule-based and machine learning-based methods can be combined to form better performing hybrid approaches for NER. NER tasks are non-trivial as they require considerable amount of time and effort by domain experts, either by creating hand-crafted rules or creating labelled training data.

Once NEs are recognized from raw text, the next step is mapping them to a target ontology. The English language is inherently ambiguous, as the same word can often refer to different concepts in different contexts. The situation is worse in biomedicine as there is an unusually large number of potentially ambiguous or even conflicting synonyms, acronyms, hypernyms and hyponyms. For example, the medical disorder autism can be referred to as autistic disorder, Kanner's Syndrome, autistic spectrum disorder, ASD or Asperger's syndrome. However, the acronym ASD in biomedicine can also refer to acute stress disorder (another medical disorder), anti-seizure drug (a medication), Aspartate-semialdehyde dehydrogenase (an enzyme), atrial septal defect (a medical condition), and possibly many other concepts in specific subdomains. Therefore, extracted NEs need to be normalized, converting from a surface form to a canonical form, and disambiguated if there is more than one matching concept in a reference ontology.

The next step in information extraction is determining the relationships between pairs of named entities. In situations where reference ontologies exist and the relationships between concepts in the ontology are well-defined, then the relationships between NEs can be easily derived from the relationships between concepts in the ontology. In cases where there are no reference ontologies, relationships can be predicted from concept co-occurrence within the text and their syntactic dependencies, semantic roles, or frequency of co-occurring terms. Extracted NEs and their relationships represent assertions, facts or knowledge distilled from text. This kind of knowledge can be represented using the Predicate Argument Structure (PAS), which defines the basic semantic unit of actions. For example, knowledge extracted from the sentence "ATP synthase converts ADP to ATP." can be represented as "convert([ADP], [ATP], [ATP synthase])" in PAS using the predefined structure "action([source], [product], [enzyme])". Knowledge represented in PAS can be used for further inference and in generating candidate answers for question answering. In the open domain, the FrameNet project [11] represents each

event using a semantic frame. This approach captures the type of the event, the entities participating in the event and their relationships to each other.

2.1.5 Evaluation Metrics

In machine learning, we use a special scoring structure called a Confusion Matrix to evaluate prediction results. A Confusion matrix assigns predictions to various categories, based on the actual label and the predicted label generated from a prediction algorithm.

Table 1 shows a confusion matrix defining True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). In machine learning, *Accuracy* is often used to measure the fraction of correct predictions among all predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

Accuracy places equal importance on True-Positives and True-Negatives, so a system with high accuracy can be accurate in making True-Positives predictions, True-Negative prediction, or both.

		Predicted Label	
		Positive	Negative
Instance Label	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 1: Confusion Matrix showing the evaluation metric for prediction results. TP denotes the number of True-Positives, FP denotes the number of False-Positives and FN denotes the number of False-Negatives.

To measure the quality of positive predictions a system predicts, we can use Precision (P), Recall (R), and F-measure (F). Precision, recall, and F-measure are often used as evaluation metrics in machine learning and text mining, but they may have different meanings in different contexts. In Information retrieval, precision is the fraction of retrieved documents that are relevant, and recall is the fraction of relevant documents that are actually retrieved. In supervised classification, precision is the fraction of data objects with correctly predicted labels, and recall is the fraction of data objects predicted with a certain class label among all data objects labelled with that same class label. F-measure combines precision and recall into a single score and reflects a system's overall performance. Precision, recall, and F-measures [37] are defined as follows:

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F-measure (F)} = \frac{2 \times \text{P} \times \text{R}}{\text{P} + \text{R}}$$

where P stands for precision, R for recall, F for F-measure.

When evaluating a system's performance in multiple runs of classifications or retrieval results for multiple queries, we use the notion of Average Precision (AP), Average Recall (AR), and Average F-measure (AF) by simply taking the arithmetic means of precision, recall, and F-measure values across multiple classification or query retrieval runs.

$$\text{Average Precision (AP)} = \frac{1}{n} \sum_i^n P_i$$

$$\text{Average Recall (AR)} = \frac{1}{n} \sum_i^n R_i$$

$$\text{Average F-measure (AF)} = \frac{1}{n} \sum_i^n F_i$$

Where n is the number of classification or query retrieval runs.

In information retrieval, we need to further discriminate systems based on both the content and the order of the retrieval results. This is because information retrieval systems, like search engines, need to rank retrieved documents in an ordered list. For example, two search engines returning the same collection of documents in a different order would show the same accuracy, precision, recall, and F-measure. However, the search engine that ranks more relevant documents higher in its results is superior to the one that shows more irrelevant documents at the top of its list. Therefore, we need to introduce the notion of Non-Interpolated Average Precision (NAP), Mean Average Precision (MAP) and Geometric Mean Average Precision (GMAP) [88, 89].

$$\text{Noninterpolated Average Precision (NAP)} = \frac{\sum_{r=1}^{\|L\|} P(r) \times \text{rel}(r)}{\|L_R\|}$$

Non-interpolated Average Precision considers both the content and the order of retrieved results. $\|L\|$ is the number of retrieved documents, and $\|L_R\|$ is the number of relevant documents in

$\|L\|$. P_r is the precision of the first r retrieved documents (fraction of relevant documents among the first r retrieved documents), and $rel(r)$ is a boolean function which equals to 1 if the r document is relevant and 0 otherwise. In this definition, NAP weights each relevant retrieved document by its ranking in the ordered list of results.

To evaluate the performance of a system over multiple runs or retrieval queries, we can average over all non-interpolated averaged precision values using either an arithmetic mean or a geometric mean as follows.

$$\text{Mean Averaged Precision (MAP)} = \frac{1}{n} \sum_{i=1}^n NAP_i$$

$$\text{Geometric Mean Average Precision (GMAP)} = \sqrt[n]{\prod_{i=1}^n (AP_i + \epsilon)}$$

In the definition of MAP and GMAP, n is the number of documents in the retrieved list and ϵ is a small value that adds to NAP_i to avoid zeroing the product of NAP_i on queries retrieving entirely irrelevant results. As we can see in these definitions, GMAP places more emphasis on retrieval results with low average precision and an information retrieval system's overall performance.

2.2 Related Work

Building a biomedical question answering system requires solutions to following challenges: 1) Document Retrieval, the retrieval of relevant documents in a large document collection; 2) Named Entity Recognition, the recognition and normalization of biomedical entities mentioned in raw text; 3) Ontology Mapping, mapping named entities to a target ontology; 4) Relation Extraction, the extraction of entity relations; and finally 5) Question Answering engines, which analyze question types, generating, scoring, and ranking candidate answers, and synthesizing the final natural language answers. This section discusses a number of related works or previously published examples for each of the aforementioned challenges.

2.2.1 Biomedical Thesaurus, Lexicons, and Ontologies

Over the past decade, much effort has been directed at curating domain specific thesauruses (or thesauri), lexicons and ontologies for biomedicine. Thesauruses typically provide names of biomedical entities, their synonyms and acronyms. Lexicons provide word senses and categorize terms into a hierarchy. Ontologies specify entities, their attributes and relationships with other entities in the same or different domains of interest. Many biomedical thesauri, lexicons, and ontologies exist, each of them serving different purposes. Here I will describe a few ontologies, thesauri and lexicons that are most relevant to this research, including Gene Ontology, MeSH, UMLS, and BioLexicon.

Gene Ontology (GO) [6] is one example of an ontology that provides a controlled vocabulary to describe gene product characteristics. The three major taxonomies in GO are cellular components, molecular functions, and biological process. Medical Subject Headings (MeSH) [81] is another example of an ontology that provides a controlled vocabulary to index biomedical publications for effective retrieval in the PubMed search engine. Unified Medical Language System (UMLS) [14] is a meta-thesaurus combining medical terminologies from SNOMED CT, MeSH, Gene Ontology, OMIM, and several other databases, for use in clinical text mining applications. The PolySearch thesauri [16, 17] are a collection of thesauri that contains comprehensive dictionaries of gene, protein, organ, tissue, subcellular compartments,

diseases, drugs, and metabolites extracted from various high quality knowledge bases and ontologies. The joint chemical dictionary (Jochem) [39] is a dictionary of chemical names for the identification of drugs and metabolites in text. BioLexicon [87] is a biological lexicon that provides a dictionary of terminologies extracted from large public bioinformatics data resources, along with their surface form variations and frequency counts calculated from MEDLINE abstracts. As essentially all biomedical literature is written in English text, open domain linguistic resources such as WordNet [62], VerbNet [83] can also be used to mine biomedical text. WordNet [62] is a lexical database of the English language. WordNet contains nouns, verbs, adjectives, and adverbs organized in collection of cognitive units (synsets). Each synset contains a set of synonyms (interlinked by their semantic relations) expressing a distinct concept. VerbNet [83] is a similar lexical database to WordNet, but it focuses on verbs and their semantic relations found in the English dictionary.

2.2.2 Document Retrieval

Document retrieval has many applications in QA. For instance, document retrieval can be used to retrieve relevant MEDLINE abstracts among millions of raw text documents, or it can be used to retrieve relevant data in a knowledgebase (e.g. UniProtKB), or relevant concepts in an ontology. PubMed is the primary tool for document retrieval for biomedical literature [66, 67]. It is part of NCBI's Entrez retrieval system and it provides efficient search interface to more than 20 million MEDLINE publications. As PubMed provides a public API, numerous other document retrieval systems have been developed based on PubMed to facilitate better result ranking, easier document navigation, and improved information digestion.

Here I will highlight three online MEDLINE/document retrieval systems: PolySearch, GOPubMed and EBIMed. PolySearch [16, 17] supports queries of the form: "given X, find all Ys", where X and Y could be diseases, tissues, cell compartments, gene/protein names, SNPs, mutations, drugs and metabolites. Results are ranked by biomedical entities, and supporting evidence are scored by the frequency of concept co-occurrence. PolySearch provides an efficient way to formulate hypothesis for discovering hidden relations between biomedical entities. GOPubMed [22] is a knowledge-based search engine for MEDLINE citations. GOPubMed

recognizes Gene Ontology (GO) terms mentioned in MEDLINE abstracts and labels text sections with GO terms. By indexing MEDLINE using GO terms, GOPubMed users can navigate MEDLINE through GO or UMLS concepts, instead of generic MeSH indexing. EBIMed [79] searches MEDLINE through user defined Boolean queries and digests the returned abstracts by recognizing gene/protein names, GO annotations, drugs, and species names. EBIMed then extracts entity relationships from the search results. Many other interesting document retrieval systems exist and many of these are described in more detail in a recent survey by Lu *et al.* [54].

Many structured biomedical databases provide application programming interfaces, or APIs, that allow text mining programs to access database content over the Internet. As information about a single biomedical entity may be scattered in many databases, there is a need for effective data capture and consolidation from multiple databases. BioSpider [49] is an example of a program that was developed to address this issue. Given a search query term, BioSpider crawls multiple biomedical databases, then fetches useful information regarding single biomedical entities. Similar to PolySearch, BioSpider also retrieves entity relations. However, in contrast to PolySearch, BioSpider only retrieves existing relations as they are specified in a reference database. DataWrangler [45] is a recently developed program for automated aggregation of chemical compounds, proteins, reactions, and pathway annotations across multiple database. In contrast to PolySearch, DataWrangler focuses on finding annotations for a compound by searching protein, reaction, and pathway annotation databases.

2.2.3 Named Entity Recognition

Named entity recognition (NER) involves the extraction of terms denoting real world entities, such as the names of people and places from raw text. The major approaches used in NER are lexicon-based, rule-based, and statistics-based. Lexicon-based methods rely on term dictionaries and thesauruses for exact or approximate term matching. Rule-based method exploits hand-crafted or machine-learned rules, usually expressed in the form of regular expressions, to identify specific text string patterns that extract the terms of interest. Statistical methods learn classification rules by training on a dataset through statistical machine learning techniques; these methods then classify novel terms to their categories. Because different NER

methods have different strengths and weaknesses, there are now several NER systems that combine all three types of NER approaches to improve performance.

In the biomedical domain, NER tasks typically involve the recognition of genes, proteins, diseases, drugs, and chemicals from raw text. In the category of lexicon-based methods, Gerner *et al.* described LINNAEUS [32], a species name identification system using the species names from the NCBI taxonomy database as their base name dictionary. LINNAEUS exploits hand-crafted rules to resolve name variants and abbreviations. As an example of a rule-based NER method, Narayanaswamy *et al.* [65] presented a method to recognize gene/protein and chemical names using a set of hand-crafted rules, and then categorized the extracted named entities based on surrounding keywords. Both lexicon-based and rule-based approaches obtain high precision but low recall if a novel term is not captured in the lexicon or expressions containing the key term do not fit any matching rules. On the other hand, statistical methods are capable of recognizing novel terms with machine-learned classifiers. For example, Akella *et al.* presented NetiNeti [1], a statistical NER system to recognize novel species names and species name misspellings using machine-learned classifiers. Given a paragraph of text, NetiNeti generates trigrams, bigrams, and unigrams as species name candidates, and then classifies each N-gram as being a species name (or not) using a Naive Bayes classifier.

2.2.4 Ontology Matching

In many text mining scenarios, extracted NEs need to be mapped to concepts in a target ontology. This concept mapping step effectively disambiguates the extracted concept, connecting an entity name to a concept in an ontology. This defines its meaning and relationship to other entities in the same ontology. Concept mapping can be formulated as a document retrieval problem, where the extracted NE is treated as a search query and the target ontology as a document (concept) collection. For example, Kim *et al.* [48] showed how it was possible to map sentences to UMLS concepts using an unsupervised information retrieval model and clustering. They retrieved concepts in UMLS as relevant documents to a given sentence (the search query) and selected representative concepts from concept clusters. Concept mapping can also be formulated as an information extraction problem, where entities are matched to concepts having

the highest degree of lexical or semantic similarity. For example, GoPubMed [22] recognizes terms in MEDLINE corresponding to concepts in the Gene Ontology (GO) collection. Noting that GO terms, if mentioned in the text, seldom occur as they appear in the Gene Ontology, GoPubMed maps different English expressions to GO terms using approximate string matching. In particular, the most general text token in a given expression is used to retrieve relevant GO terms, and these GO terms are progressively refined using more specific terms in the expression [22].

Mapping extracted NEs to a target ontology is a data integration task that is associated with a certain degree of uncertainty. In particular, the data sources where NEs are extracted may not be perfect, and the mappings between extracted NEs and the target ontology may not be certain. Dong *et al.* [23] address this “data integration with uncertainty” problem by introducing a probabilistic schema-mapping framework, which attaches probabilities to each named entity to concept mapping tuple. With a probability attached to each mapping tuple, the top K answer tuples are retrieved to answer an input query. Dong *et al.* shows that this probabilistic schema mapping framework is able to handle uncertainty in multiple levels including underlying data for extracting NEs, mapping schema between NEs and target ontology, and also input queries.

Once terms are mapped to a target ontology, it is possible to assess the semantic similarities between terms using the relationships between their corresponding concepts. Ontology-based semantic similarity can be edge-based or node-based. Edge-based semantic similarity counts the number of edges between two concepts in an ontology, and node-based semantic similarity examines ancestors and children of both concepts and then calculates the similarity based on the information content (IC) of these nodes [74].

2.2.5 Relationship Extraction

It is possible to extract or predict relationships between pairs of named entities co-located in raw text from their textual context. Relationship extraction is usually built on top of the results of shallow syntactic parsing and semantic role labeling. More specifically, extraction and conversion rules are used to convert the parsed sentence into “relation tuples” or predicates.

Machine learning methods can be used to score and select the final set of extracted tuples. In the biomedical domain, there has been a great deal of attention focused on extracting protein-protein interactions (PPI), as highlighted by the BioCreative challenges [50]. The BioCreative challenges were competitions designed motivate researchers to improve extraction accuracy on PPI. This led to many publications on mining PPI from text [102]. Recently, the research community has turned their attention to extracting and predicting other relation types, such as disease-drug, and food-disease relations [101]. Relation extraction for biomedical entities will likely remain as one of the most active topics in biomedical text mining for the next few years.

2.2.6 Question Answering

All of the aforementioned techniques are necessary to create and curate suitable knowledgebases for question answering. However, developing an effective question answering engine is equally important, as specific question answering techniques are needed to generate sensible answers for questions typically posed by users. Even though methodologies and applications for different QA systems may vary, the underlying architecture is generally quite similar. Figure 9 shows the general architecture of a stereotypical QA system, as described by Athenikos *et al.* [7]. In the question processing phase, a QA system first identifies the question type and the answer type (question analysis) from the question posed by users. It then converts the posed question into a well-formatted search query (query formulation). The converted query is then searched against a knowledge base or document collection for relevant documents, passages, or database entries (document retrieval). The QA system then generates candidate answers based on the search results (candidate generation). It also gathers evidence for each candidate through further searches and then scores and ranks them (candidate scoring and ranking). Finally, the most probable answer candidate is chosen based on ranking and other filtering criteria, and used to synthesize the final answer with evidence (answer synthesis).

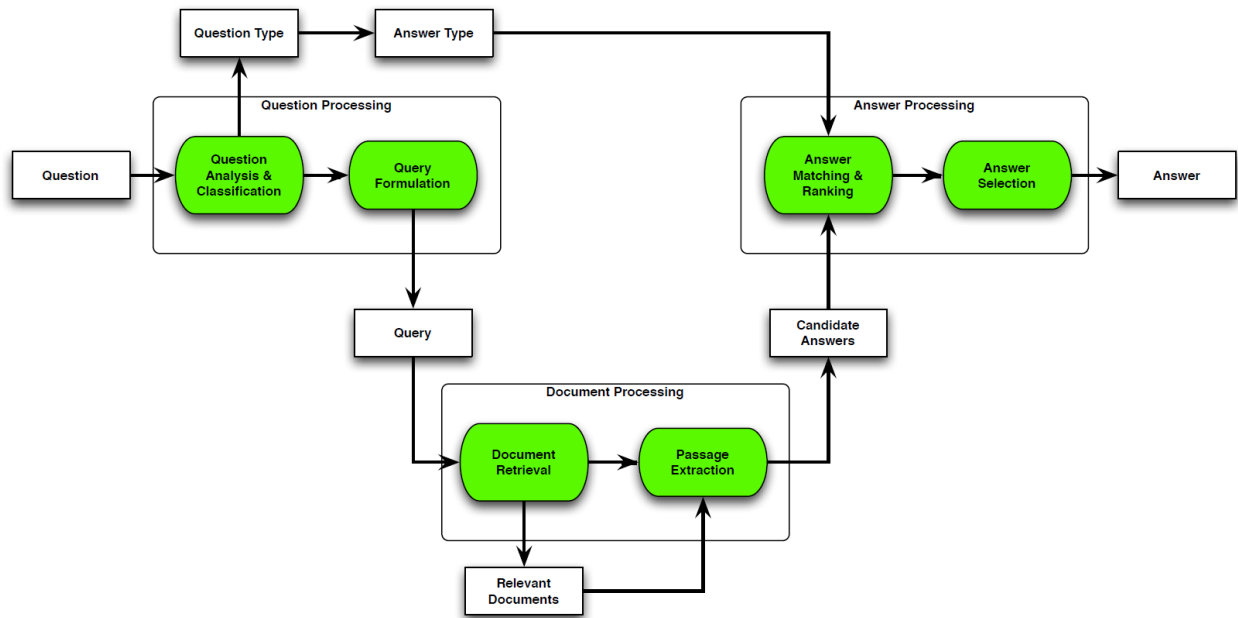


Figure 9: General architecture of a QA system. This figure is based on a figure found in “Athenikos, S.J., and Han, H. (2010) Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1-24, July 2010.”

Athenikos *et al.* categorized open domain QA systems into three categories: 1) semantic-based; 2) inference-based, and 3) logic-based systems. Semantic-based QA systems exploit lexicon-semantic information extracted from documents. Inference-based QA systems make use of inference rules to make inferences based on question clues and existing assertions. Logic-based QA systems employ explicit logic formulations and theorem-proving techniques to answer questions usually posed as logical statements that can be proved or disproved [7]. In recent years, many different QA systems have emerged to answer practical questions in both open and specialized domains. IBM’s Watson program, with its QA engine called DeepQA, defeated human contestants in the famous quiz show *Jeopardy!* [36]. DeepQA is mostly a semantic-based QA system, but it is also capable of generating answer candidates using inference and logic. DeepQA relies on the PRISMATIC knowledge base, which contains a set of semantic units called “frames” capturing entities and their relations extracted from a free text corpus like the

Wikipedia [24]. DeepQA determines the type of answer, or Lexical answer types (LATs), from the given question. DeepQA then uses multiple approaches to generate candidate answers through its extensive knowledgebase, its database, and various web searches. The supporting evidence retrieval unit of DeepQA then retrieves text passages containing a candidate answer and relates them back to the original question. Candidate answers with evidence passages that are more relevant to the original question receive a higher score and a better ranking. In addition to its exceptional question answering capabilities, DeepQA also has a module that determines when to bet, which question to bet on, and how much to bet in order to maximize its final game score. I will not go into the details here as the betting module is irrelevant to correctly answering a question that has been posed, and thus not particularly relevant to the proposed research.

Building on the success of *Jeopardy!*, IBM is currently adapting DeepQA as a clinical support system to assist health care providers in making treatment decisions [10]. Over the past decade, another company called SRI international conducted yet another large-scale question answering project: Project Halo [25]. This project aims at creating a “digital Aristotle” that assists students to learn and scientists to perform their daily research. In contrast to DeepQA, project Halo takes an inference-based QA approach. Project Halo employs dozens of trained knowledge engineers and domain experts to encode textbook knowledge from textbooks as machine readable ontologies and inference rules, which would enable intricate inference and answer explanation. Project Halo has been reported to be able to answer questions at the Advanced Placement test level. Project Halo continues to advance towards the goal of answering college level and advanced research level questions. Despite their success in answering open domain questions, both DeepQA and Project Halo are not yet publicly available. In contrast, True Knowledge [91] and Wolfram Alpha [99], two commercial open domain QA systems, are available over the Internet. True Knowledge is a semantic-based QA system and it uses a similar approach as DeepQA. Methodologies and implementation details of Wolfram Alpha are unclear due to the limited number of publications for this commercial system.

In the biomedical domain, QA systems can be classified roughly into two categories: clinical QA systems and biological QA systems. Many recent biomedical QA systems focus on providing support to disease diagnosis and clinical decision making; therefore, they fall into the clinical QA category. Biological QA systems, on the other hand, are more focused on answering

broad questions posed in biological research that are also interesting to the medical community. To the best of our knowledge, very few biological QA systems exist and none of them are publicly available or web accessible. Takahashi *et al.* [84] proposed to build a semantic-based biomedical QA system in 2004, but no implementation details are provided. Gu *et al.* [34] built BioSquash, a QA-oriented multi-document summarization system, which summarizes relevant documents for a given question and presents the summarization as an answer. The source code for BioSquash is available upon request but no web interface is provided for the public. Anwar *et al.* [2] proposed a framework called BioPathQA that specialized in answering user queries about biological pathways. BioPathQA uses Petri Nets to encode biological pathways and it also supports pathway simulations. BioPathQA requires that the user compose queries in BioPathQA's specialized logical query language, and does not support natural language queries or provide textual answers. BioPathQA has been proposed as a framework and no public server or API is available to serve the general public.

With the emergence of several publicly accessible biomedical QA systems there is now a growing need to provide a common platform for evaluating biomedical QA systems. This need is what has motivated the series of BioASQ challenge. BioASQ (<http://bioasq.org>) is a semantic question answering competition with two well-defined shared tasks. Task A challenges participants to automatically index novel MEDLINE abstracts with MeSH tags; Task B challenges participants to annotate given natural language questions with relevant articles, text snippets, and RDF triples from designated document and concept repositories (Phase A), and eventually return an “exact” and “ideal” answer in natural language (Phase B). Participants are allowed to process a challenge question set and submit answers within 24 hours. Submission results are evaluated both automatically and manually by a panel of biomedical experts. Much more detail about the BioASQ challenge is provided in Tsatsaronis *et al.* [88, 89].

In this thesis, I will present both a framework and a prototype, web-based biomedical QA system called BioQA. BioQA falls into the semantic based category of biological QA systems. This is because BioQA relies on information extracted from databases and text snippets from relevant sentences to synthesize its answers. As a biological QA system, BioQA focuses on answering general questions that arise in the biological and biomedical domains, and not clinical questions specific to medicine. I developed BioQA to answer biomedical questions posed by

medical researchers, life scientists, life science students and the general public. Over the next two chapters I will describe PolySearch2, a core building block in BioQA, as well as BioQA itself.

3. PolySerach2: A Text Mining Framework

A critical task in biomedical question answering and biomedical text mining is the discovery of potential associations between various types of biomedical entities or subjects. This chapter introduces PolySearch2¹ (<http://polysearch.ca>), an online text-mining system for identifying relationships between biomedical entities such as human diseases, genes, SNPs, proteins, drugs, metabolites, toxins, metabolic pathways, organs, tissues, subcellular organelles, positive health effects, negative health effects, drug actions, Gene Ontology terms, MeSH terms, ICD-10 medical codes, biological taxonomies and chemical taxonomies. PolySearch2 supports a generalized “Given X, find all associated Ys” query, where X and Y can be selected from the aforementioned biomedical entities. In this chapter, we introduce the PolySearch algorithm, then we describe the PolySearch2 web server, and its improvements over the original PolySearch system. Finally, we evaluate the performance of PolySearch2 versus the original PolySearch system and discuss limitations and future works.

3.1 Introduction

Keeping pace with the rapidly growing body of biomedical literature is proving to be almost impossible. According to a study by Baasiri *et al.* [9] a researcher would have to scan 130 different journals and read 27 papers per day to follow a single disease, such as breast cancer. A more recent study by Lu *et al.* [54] showed that the total number of references in MEDLINE, a central repository for scientific articles in the biomedical domain, now exceeds 25 million and is growing at more than 4% each year. It is also evident that a considerable amount of useful biological or biomedical knowledge is essentially buried in the form of free text, waiting to be found and transformed into more accessible formats. Swanson referred to such phenomena as “undiscovered public knowledge” [13]. The enormous challenges associated with keeping up or

¹ Portions of this chapter were published in Nucleic Acids Research under the reference of “Liu, Y., Liang, Y., Wishart, D. (2015) *PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more.* Nucleic Acids Res. 2015 Jul 1;43(W1):W535-42. doi: 10.1093/nar/gkv383. on April 29, 2015.

digging through this undiscovered public knowledge, especially in the area of biomedical knowledge, has led to the development of a number of text mining tools aimed at supporting biomedical text extraction, fact finding and text summarization. Some of the better-known or more widely used tools include EBIMed [79], CiteXplore [59] and GoPubMed [22]. Their intent has been to help life science researchers keep pace with the exploding quantity of scientific literature and to facilitate the discovery or re-discovery of important facts or unexpected associations. The latter task of “association discovery” is of particular interest and is typified by queries such as “Find all genes that are associated with a given disease” or “Find all drugs that target a specific protein” or “Find all toxins that damage a specific tissue”. These are queries that are either not easily performed or impossible to perform through a regular PubMed search. To address this task of association discovery we previously developed a relationship or association mining tool called PolySearch [16, 17]. PolySearch was one of the first web-enabled text mining tools to support comprehensive and associative text searches of PubMed abstracts. Specifically, the original version of PolySearch supports 'Given X, find all associated Y's' types of queries, where X and Y are biomedical terms pertaining to human health and biology. X's can be genes, SNPs, proteins, diseases, drugs, metabolites, pathways, tissues, organs, and sub-cellular organelles or structures, or a general text keyword; while Y's can be any or all types mentioned above. PolySearch's search strategy is based on a critical assumption that the greater the frequency with which an X and Y association occurs within a collection of sentences or database records, the more significant the association is likely to be. For example, if Bisphenol A (BPA) is mentioned 615 times in PubMed as being associated with breast cancer, and only 8 times being associated with colon cancer, then one is more likely to have higher confidence in the potential BPA-breast cancer association over the BPA-colon cancer association.

PolySearch has proven to be both popular and effective with >20,000 users and >150 citations. It has also served as an important text-mining and annotation system for the curation of a number of metabolomics databases including DrugBank [51], HMDB [98], T3DB [97], YMDB [44], and ECMDB [35]. PolySearch has also been used to assist in disease gene discovery [64] [29], protein-protein interaction studies [86, 78], microarray data analysis [26], metabolome annotation [35, 44, 77, 98], biomarker discovery [73], as well as in building and assessing other biomedical text-mining tools [43, 90]. PolySearch has also been featured in many published

biomedical text-mining surveys and tutorials [27, 54, 64]. However, a key limitation with PolySearch has been the long search times (2-3 minutes), its limited synonym set (thesauri) and its relatively small number of searchable databases. Indeed, since its introduction in 2008 many other searchable databases and electronic free-text collections have become available and many technological improvements in web interface design, text searching and text mining have taken place. Likewise, many PolySearch users have requested more search options such as MeSH terms, adverse health effects, animal taxonomies, medical terms, Gene Ontology and chemical ontology terms. In response to these requests and many ongoing technical developments we have created a second, much improved version of PolySearch (PolySearch2, available online at <http://polysearch.ca>). This faster (up to 25X) and much improved version now has a far more robust underlying framework. It also includes a much larger collection of databases (20 vs. 7), search terms pairs (308 vs. 66), thesauri (20 vs. 9), terms (1,131,328 vs. 57,706) and synonyms (2,848,936 vs. 353,862) as well as a substantially improved and modernized interface and its underlying search algorithms. We have also upgraded the physical server to further improve its performance. A complete description of the new, updated PolySearch2 server follows.

3.2 The PolySearch algorithm

PolySearch supports 'Given X, find all associated Y's' types of queries, where X and Y are biomedical terms pertaining to human health and biology. This section describes the PolySearch algorithm, which is fundamental to both the original PolySearch [16, 17] and PolySearch2. In this section, “*PolySearch*” refers to the PolySearch algorithm and not specific to the original PolySearch web server.

PolySearch's search strategy is based on an assumption that the greater the frequency with which an X and Y association occurs within a collection of sentences or database records, the more significant the association is likely to be. PolySearch uses a text ranking scheme to score relevant sentences containing the query and other relevant biomedical terms. The text ranking scheme assigns relevancy scores to pertinent sentences and text paragraphs according to their “strength” as supporting evidence for potential associations. Given a query term, PolySearch first retrieves relevant documents from document collections and breaks each

document into individual sentences. PolySearch then scans each sentence and tries to find the query term, the association words, and related thesaurus words derived from the query and association words. Each relevant sentence, based on the frequency and placement of query, association, and/or thesaurus terms, is classified into four categories [R1 (best), R2, R3, R4 (worst)], in decreasing order of relevancy to the search query. An R4 sentence only contains one or more thesaurus terms. Typically, R4 sentences provide baseline statistics of the occurrence frequency of thesaurus terms in documents relevant to the query term. An R3 sentence contains at least one thesaurus term as well as the query term. As a general rule, R3 sentences represent stronger evidence for co-occurrence between the query term and relevant thesaurus terms. An R2 sentence satisfies all the constraints of an R3 sentence, as well as containing at least one of the association words. R2 sentences represent even stronger evidence for co-occurrence between query and thesaurus terms. Finally, an R1 sentence is a sentence that it satisfies all constraints of an R2 sentence, as well as passing specific pattern recognition criteria. R1 sentences represent the strongest evidence PolySearch can find among relevant documents to support the association assertion between query and thesaurus terms.

PolySearch identifies R1 sentences for three main types of patterns: 1) “Query Term-Association Word-Thesaurus Term”, or a QAT pattern. e.g. “A interacts with B”; 2) “Query Term-Thesaurus Term-Association Word”, or a QTA pattern. e.g. “A B interaction”, and 3) “Association Word-Query Term-Thesaurus Term”, or an AQT pattern. e.g. “Interaction between A and B”. Each pattern also imposes further rules to limit the number of words (or tokens) within the sentence fragment matching a pattern, as well as between Association words, Query terms, and Thesaurus terms. For example, in a compact QAT pattern, the number of tokens matching the pattern must be less than 10. When overlapping patterns are present, the most compact pattern will be recognized and recorded. For instance, an R2 sentence matching a specific pattern is promoted to an R1 sentence. For more implementation details on PolySearch’s pattern recognition rules, please refer to Cheng *et al.* [16]. Once relevant sentences are assigned to R1, R2, R3, and R4 categories they are then scored. Each sentence receives points based on pre-defined scoring scheme according to the document source. For example, in PolySearch2, an irrelevant MEDLINE abstract sentence receives 0 points, an R4 sentence receives 1 point, an R3 sentence receives 5 points, an R2 sentence receives 25 points, and an R1 sentence receives 50

points. This scoring scheme can be different for different sentences identified from different document collections and databases. For example, PolySearch2 assigns twice as many points to relevant sentences in database records than sentences found in free-text articles. In this case, an R1 sentence receives 2 points, an R3 sentence receives 10 points, an R2 sentence receives 50 points, and an R4 sentence receives 100 points. This scoring algorithm places heavier weights on database records than free-text documents, to show that more trust is placed on database records than free-text documents as database records have gone through a curation process and are therefore more trustworthy as source of supporting evidence. The total score of R1, R2, R3, R4 sentences found in all relevant documents and database entries for a specific Query term and Thesaurus term pair is the overall PolySearch Relevancy Score for the pair. PolySearch calculates Relevancy Scores for every Query and Thesaurus term pair and converts the raw Relevancy Score for each pair to a standardized Z-score statistic. The conversion from the raw Relevancy Score to a standardized Z-score statistic is necessary, as the raw Relevancy Score does not consider background probability for a term pair to co-occur in relevant documents by chance. The Z-score statistic is a measure for relative importance of a particular term pair among all other relevant term pairs. The formula for converting the Relevancy Score to a Z-score is shown below:

$$Z_{x,y} = \frac{R_{x,y} - \bar{R}}{\sigma}$$

In this formula, $Z_{x,y}$ is the standardized Z-score statistic for the Relevancy Score $R_{x,y}$ for term pair (x, y). \bar{R} is the average Relevancy Score and σ is the standard deviation of the Relevancy Scores among all term pairs. Finally, each Query-Thesaurus term pair is ranked based on their standardized Z-score. Term pairs with higher positive Z-scores correspond to stronger evidence for an association, as the observed co-occurrence is less likely due to chance. Term pairs with zero or negative Z-scores correspond to weak or no evidence for association, as the observed co-occurrence is more likely due to chance. Pairs with negative Z-scores are removed from the final results.

3.3 Improvements and Enhancements in PolySearch2

PolySearch2 (<http://polysearch.ca>) features a number of improvements and enhancements including: 1) algorithmic improvements; 2) an improved graphical interface and the implementation of modernized web technology; 3) significant database and text search enhancements; 4) substantially expanded synonym sets and thesaurus types; and 5) improved caching and updating. These changes have also led to substantial performance improvements relative to the earlier version of PolySearch. Details regarding these changes and improvements are described below.

3.3.1 Algorithmic Improvements

PolySearch2 incorporates a number of algorithmic improvements aimed at strengthening the scoring, ranking, and selection of association term candidates. These include: 1) a new “tightness measure” to further discriminate association patterns, 2) a “weight boost” for database records to favor explicit database associations over free-text articles, 3) a larger collection of system filter words, and 4) a filter to remove borderline associations.

PolySearch2 now uses a “tightness measure” to reward more proximal word co-occurrences and penalize more distant word co-occurrences. Just as in the original version, PolySearch2 assigns relevant sentences into four categories (R1 [best], R2, R3, and R4 [worst]) based on a relevancy score as derived from the search query and the matched co-occurrence patterns. However, PolySearch2 now measures the word span between matched co-occurrence patterns found in a relevant sentence. In particular, it assigns higher relevancy scores to tighter patterns with fewer words separating the query term and target term(s), and lower relevancy scores to more relaxed patterns with a larger word span between the query term and the target term(s). An example of an R1 sentence with a tight co-occurrence pattern could be “Exposure to bisphenol A (BPA) increases the risk of breast neoplasms”, while an example R1 sentence with relaxed co-occurrence pattern could be “Bisphenol A may play a role in gene regulation pathways that are potentially related to the onset and development of breast cancer.” We found

this tightness measure improves the scoring of co-occurrences and enhances PolySearch2's ability to distinguish genuine associations from incidental co-occurrences that arise by chance.

Unlike the original version of PolySearch, PolySearch2 now assigns greater weight to relevant database records than free-text articles. It has been previously shown [16, 17] that including database records in the search process consistently improves association accuracy. Generally, database records contain high quality, well-structured and carefully curated knowledge whereas free-text articles generally contain more ambiguous, implicit knowledge. Therefore, it stands to reason that database records should be assigned higher credibility than text articles. However, given the sheer volume of biomedical publications and the relatively small number of high quality biomedical databases, one is more likely to find relevant free-text articles than database records. To counter this bias, PolySearch2 applies an empirically determined “weight boost” to the information it finds in database records and assigns greater relevancy scores to relevant database records than free-text articles. The “weight boost” reflects the difference in credibility associated with database records compared to sentences in free-text articles.

PolySearch2 also incorporates a more extensive collection of “system filter” words than the original version of the program (29,718 filter words vs. 7,011 filter words). In particular, PolySearch2 now recognizes co-occurrence patterns more consistently thanks to this larger, more extensive collection of filter words. System filter words are essentially words that signify a strong association. For example, the word “catalyzes” in “Enzyme X catalyzes reaction Y” indicates a strong association between Enzyme X and reaction Y. The new and improved set of filter words were initially mined from the entire collection of MEDLINE abstracts using Natural Language Processing techniques. In creating PolySearch2's list of system filter words, we tagged the occurrence of all biomedical entities in the current collection of MEDLINE abstracts, extracted text flanking each pair of co-occurrence entities, and classified the flanking text according to the co-occurring entity types. We then built N-gram models for common verbs, adjectives, adverbs and phrases present in the flanking text for each pair of co-occurrence entity types. The list was carefully assessed and manually curated to produce the final filter word set. This collection of system filter words helps PolySearch2 recognize strong associations from

mere co-occurrences. It also allows it to perform consistently better at recognizing term associations than the original version of PolySearch.

The final algorithmic enhancement to PolySearch2 involved the application of a more stringent cut-off to boost precision at the cost of sacrificing a small degree of recall (i.e. the precision-recall trade-off). Associations discovered in PolySearch2 are ranked and sorted using Z-scores calculated from PolySearch2's raw relevancy score (See [16, 17]). Associations with average relevancy scores are assigned zero Z-scores, as they represent borderline or marginal associations derived from a particular search. PolySearch2 now removes associations with zero Z-scores to boost its precision. This is done at the risk of removing a small number of possible genuine associations. For users concerned about the emphasis of recall over precision in their results, PolySearch2 also provides an option to include borderline cases (or 'zero Z-score' associations).

3.3.2 Graphical Interface and Web Implementation

PolySearch2 (<http://polysearch.ca>) features a completely re-designed web interface. Figure 10 to Figure 14 show screenshots of various pages from PolySearch2's new web interface. Figure 10 shows the query submission page where users can initialize a search query. As with the original PolySearch, PolySearch2 still supports a 'Given X find all associated Y's' type of query. Users can initialize a search by selecting the desired type of X (query term) and Y (target term) from pull-down menus and enter a search query keyword. At this point user can submit a "Quick Search" request (Figure 10) or further configure the search using "Advanced Options" (Figure 11). Both of these features are new to PolySearch2. The Quick Search option will direct PolySearch2 to search previously computed cache results or to mine associations from the top 2000 relevant articles or database records across all text collections and databases. In the Quick Search, PolySearch2 automatically generates a synonym list (from the PolySearch2 thesauri) and proceeds with its regular searching, sorting, scoring, annotation and display (described in detail in [17]). "Advanced Options" (Figure 11) offers a greater degree of customizability to the search. For instance, users can edit the automatically generated synonym list (from the PolySearch2 thesaurus), edit custom filter words for identifying association patterns, provide custom negation

words for filtering out sentences with negative associations, provide custom target terms to search, select or de-select source text collections and databases, indicate the number of documents to search, permit the inclusion or exclusion of hits with zero Z-scores (for higher recall), and/or provide an E-mail address for notifications.

POLYSEARCH2 Search Check Result Thesaurus Documentation About Contact Us

PolySearch2

Quick Start

To use this server:

1. Decide which type of search you wish to do (e.g. "Given Toxin Find associated Diseases")
2. Select search restraints from the pull-down menus (Given X, Find Y)
(e.g., select "Toxin" from the Given menu and "Disease" from the Find menu.)
3. Enter Query Keyword (e.g. "Bisphenol A")
4. Press "Quick Search" to start a search using default settings, OR
5. Press "Advanced Search" and follow the instructions on the advanced search page to fine tune your search
6. If you need more help or detailed explanations of the methods or databases, see the help section.

Choose your search type and enter query keyword

Given Disease **Find ALL associated** Diseases

Query Keyword Bisphenol A

Please cite:

Liu Y., Liang Y., Wishart D.S. (2015) PolySearch 2.0: A significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins, and more. Nucleic Acids Res. 2015 (Web Server Issue) Manuscript submitted.

This project is supported by the [Canadian Institutes of Health Research](#) (award #111062), [Alberta Innovates - Health Solutions](#), and by [The Metabolomics Innovation Centre \(TMIC\)](#), a nationally-funded research and core facility that supports a wide range of cutting-edge metabolomic studies. TMIC is funded by [Genome Alberta](#), [Genome British Columbia](#), and [Genome Canada](#), a not-for-profit organization that is leading Canada's national genomics strategy with \$900 million in funding from the federal government.

Figure 10: A screenshot of PolySearch2's query interface showing the PolySearch2 query submission form.

PolySearch2 - Given **Toxins** Find Associated **Diseases**

Advanced Options:

Search keyword	<input type="text" value="Bisphenol A"/>	
Automated synonym list	<input type="text" value="4,4-Isopropylidenediphenol; P, p-dihydrox"/>	
Please enter custom filter words (default is given), separate words with ";" (eg. gene; polymorphism)	<input type="text" value="accelerate;achieve;acute;address;affe ct;alter;ameliorate;amplify;analyse;ana lvze;assist.aument.benefit:chance.ch"/>	
Custom Negation Words, separate words with ";"	<input type="text" value="not observed; no evidence; not present; ir"/>	
Custom Thesaurus	<input type="text"/>	
Select one or more Corpus to search. (For faster computation, only PubMed is selected as a default)	<input checked="" type="checkbox"/> PubMed <input type="checkbox"/> PubMed Central <input type="checkbox"/> Wikipedia <input type="checkbox"/> USPTO Patent <input type="checkbox"/> NCBI Books <input type="checkbox"/> MedlinePlus	
(Optional) Select one or more database to search.	<input type="checkbox"/> DrugBank <input type="checkbox"/> HMDB <input type="checkbox"/> T3DB <input type="checkbox"/> YMDB <input type="checkbox"/> ECMDB <input type="checkbox"/> OMIM <input type="checkbox"/> UniProt - SwissProt <input type="checkbox"/> GAD <input type="checkbox"/> Gene Ontology <input type="checkbox"/> HPRD <input type="checkbox"/> DailyMed <input type="checkbox"/> KEGG Pathways <input type="checkbox"/> KEGG Reactions <input type="checkbox"/> MetaCyc	
Document Limit	<input type="text" value="2000"/>	
Show borderline associations.	<input type="checkbox"/> Show Borderline Associations	
(Optional) Please send the results to me by email.	<input type="text" value="your Email address"/>	

Figure 11: A screenshot of PolySearch2's query interface showing the advanced option page for further query refinement.

Success! Showing cache results computed on 2016-09-21 09:17:46 .

PolySearch2 - Given Diseases Bisphenol A, Find Diseases

Association Search ID 1474575198 / Given Bisphenol A

Repeat My Search with Latest Results

23 Diseases

ZScore	RScore	Entity ID	Name	Synonyms	Hits	Details
24.14	730	DID71361	Obesity	Obesity... (Read More)	51 [1, 0, 19, 31]	Details
20.65	630	DID38503	Diabetes mellitus	Diabetes mellitus; DM - Diabetes mellitus; diabetes; Diabetes mellitus (disorder); Diabetes nos... (Read More)	39 [1, 0, 17, 21]	Details
14.21	445	DID06582	breast neoplasm	breast neoplasm; Breast tumour; Neoplasm of breast; Breast Neoplasm; Tumors, Breast; Breast Tumor; Tumor of the Breast; Tumor, Breast; mammary tumor; Breast Neoplasms; Tumor of breast; Tumour of breas... (Read More)	41 [0, 0, 12, 29]	Details
6.02	210	DID55595	malignant tumoral disease	malignant tumoral disease; Malignant neoplastic disease; Cancer morphology; Malignant tumor morphology; malignant tumor; Malignant Neoplasms; neoplasm/cancer; Malignant neoplasm; Malignant tumour; Tum... (Read More)	26 [0, 0, 4, 22]	Details
5.85	205	DID49267	Prostate Neoplasms	Prostate Neoplasms; Tumor of the Prostate; Neoplasm of the Prostate; prostate neoplasm; Tumor of Prostate; Prostatic Neoplasms; Tumour of prostate; prostatic neoplasm; Prostate Tumor; PROSTATE NEOPLAS... (Read More)	17 [0, 0, 6, 11]	Details
4.98	180	DID34242	High Blood Pressures	High Blood Pressures; hyperpiesis; HBP - High blood pressure; vascular hypertension; Blood Pressure, High; Hyperpiesia; Hypertensive diseases; high blood pressure; Hypertensive disease; HT - Hypertens... (Read More)	16 [0, 0, 5, 11]	Details
3.23	130	DID37448	Metabolic syndrome	Metabolic syndrome; metabolic syndrome... (Read More)	10 [0, 0, 4, 6]	Details

Figure 12: A screenshot of PolySearch2's result display showing the PolySearch2 result overview table.

Diseases : DID06582 - breast neoplasm

Association Search ID 1474575198 / Given Bisphenol A / Found Diseases / DID06582 - breast neoplasm

ZScore	RScore	Entity Type	ID	Name	Synonyms
14.2066656954	445 - [0, 0, 12, 29]	Diseases	DID06582	breast neoplasm	breast neoplasm; Breast tumour; Neoplasm of breast; Breast Neoplasm; Tumors, Breast; Breast Tumor; Tumor of the Breast; Tumor, Breast; mammary tumor; Breast Neoplasms; Tumor of breast; Tumour of breast; Neoplasm of the Breast; Mammary Neoplasms; Breast Tumors; NEOPLASM BREAST; Neoplasm, Breast; Breast cancer; Breast cancer; Ca breast - nos; Malignant neoplasm of breast; Malignant tumor of the breast; Mammary cancer; Primary breast cancer

[Back to View Results](#)

20 MEDLINE 3 PubMed Central

RScore	Document	Snippets	Details
60 - [0, 0, 2, 2]	Effects of bisphenol A on breast cancer and its risk factors. [MEDLINE : 18843480]	<p>Yang M, Ryu JH, Jeon R, Kang D, Yoo KY (2009) Effects of bisphenol A on breast cancer and its risk factors. Archives of toxicology;Arch. Toxicol.;2009 Mar;83(3):281-5 (PMID: 18843480)</p> <ul style="list-style-type: none"> - Effects of bisphenol A on breast cancer and its risk factors. - The incidence of breast cancer in Korea has been increasing for the last two decades (1983-2005), and now, breast cancer is ranked the leading cause of cancer in Korean women. - Along with other endocrine disrupting chemicals (EDCs), bisphenol A (BPA) has been suspected as a potential risk factor for breast cancer. - Considering interactions between BPA exposure and risks of breast cancer, we suggest further enlarged biomonitoring studies of BPA to provide effective prevention against breast cancer. 	Details
40 - [0, 0, 1, 3]	Bisphenol-A induces expression of HOXC6, an estrogen-regulated homeobox-containing gene associated with breast cancer. [MEDLINE : 25725483]	<p>(2015) Bisphenol-A induces expression of HOXC6, an estrogen-regulated homeobox-containing gene associated with breast cancer. Biochimica et biophysica acta;Biochim. Biophys. Acta;2015 Jun;1849(6):697-708</p> <ul style="list-style-type: none"> - E2 induces HOXC6 expression in cultured breast cancer cells and in mammary glands of Sprague Dawley rats. - Our studies demonstrate that HOXC6, which is a critical player in mammary gland development, is upregulated in multiple cases of breast cancer, and is transcriptionally regulated by E2 and BPA, in vitro and in vivo. - We observed that HOXC6 is differentially over-expressed in breast cancer tissue. - Bisphenol-A induces expression of HOXC6, an estrogen-regulated homeobox-containing gene associated with breast cancer. 	Details

Figure 13: A screenshot of PolySearch2's result display showing the detailed result page with supporting evidence for a single association (Bisphenol A – Breast Neoplasm).

PolySearch2 - Result Details

Association Search ID 1474575198 / Given Bisphenol A / Found Diseases / DID06582 - breast neoplasm / MEDLINE 18843480

Yang M, Ryu JH, Jeon R, Kang D, Yoo KY (2009) Effects of bisphenol A on breast cancer and its risk factors. Archives of toxicology;Arch. Toxicol.;2009 Mar;83(3):281-5 (PMID: 18843480)

The incidence of breast cancer in Korea has been increasing for the last two decades (1983-2005), and now, breast cancer is ranked the leading cause of cancer in Korean women. Along with other endocrine disrupting chemicals (EDCs), Bisphenol A (BPA) has been suspected as a potential risk factor for breast cancer. We studied potential associations between BPA exposure and breast cancer risks in Korean women by performing biomonitoring of BPA among breast cancer patients and controls (N = 167). Blood samples were collected between 1994 and 1997 and kept over 10 years in a freezer under well controlled conditions. The blood BPA levels determined by HPLC/FD, ranged between LOD (0.012 microg/L) and 13.87 microg/L (mean +/- SD, 1.69 +/- 2.57 microg/L; median, 0.043 microg/L). In age-matched subjects (N = 152), there were some associations between BPA levels and risks of breast cancer, such as age at first birth and null parity. However, there were no significant differences in blood BPA levels between the cases and the controls (P = 0.42). Considering interactions between BPA exposure and risks of breast cancer, we suggest further enlarged biomonitoring studies of BPA to provide effective prevention against breast cancer.

MEDLINE 18843480 : Effects of bisphenol A on breast cancer and its risk factors.

[← Back to View Evidence](#)
[←← Back to View Results](#)
[View Article on PubMed](#)

Legend

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Query Term	Association Word	Drug, Toxin	Gene, Protein, Gene Family, Pathway, SNP	Human Metabolite, Food Metabolite, MeSH Compound	Tissue, Organ, Subcellular Localization	MeSH Term, Gene Ontology, Chemical Taxonomy, ICD-10 Code	Health Effect, Drug Effect, Adverse Effect	Disease, Species

Figure 14: A screenshot of PolySearch2's result display showing result details with the full MEDLINE abstract in highlighted and hyperlinked text.

Once a search query is submitted, the user will be redirected to a progress page where the user can bookmark the page for later visits. When a search is completed, the user will be redirected to a results overview page (Figure 12) showing the associated entities of the selected target category (or all categories if the search is against ALL target categories). In Figure 12, a screenshot listing the diseases found to be associated with the toxin Bisphenol A is shown. The resulting overview table is sorted by Z-scores in descending order, and can be sorted according to values in a certain column by clicking on the column header. The overview table lists the Z-score and PolySearch Relevancy Score (R-score) as well as the name and synonyms for each associated entity. Users can review query settings, browse through full tables in a printable format or download their results in JSON format by clicking the appropriate links on this page. Clicking on the "Details" button on each row takes users to a detailed result page (Figure 13) showing the supporting evidence in color-coded and hyperlinked sentences from each relevant article in each text collection or biomedical database. For results with MEDLINE abstracts or PubMed Central articles, there is an additional "Details" button for each row. Clicking on this specific "Details" button takes user to view the full MEDLINE abstract in highlighted and hyperlinked text (Figure 14). A result navigation bar with light grey background just below the headers of all result pages (Figure 12, Figure 13, and Figure 14) is provided for users to quickly review and navigate within the result hierarchy. These features are described in more detail on PolySearch2's Documentation web page.

In addition to the substantially modified and updated graphical user interface, PolySearch2 also underwent a complete upgrade and re-implementation of the web front-end using the latest web technology standards (HTML5 & Twitter Bootstrap). We have also upgraded the underlying physical server to further improve its performance. PolySearch2's back-end API and front-end web server are deployed on a dedicated tower server machine with 8 cores operating at 1.4GHz and multiple Solid-State Drives to facilitate rapid document retrieval and analysis. A PolySearch2 API for bulk text mining is also available upon request (with certain limitations). The architecture of PolySearch2 (see Figure 15) also allows it to easily scale up its computation across multiple machines on a computer cluster or cloud platform should further upgrades be needed. PolySearch2 uses the Model-View-Controller (MVC) design pattern: 1) the PolySearch2 Search Engine with Elasticsearch (Model layer) organizes document collections. 2)

the PolySearch2 API (Controller layer) implements the core PolySearch2 algorithms and queries the model layer for search results. 3) the PolySearch2 web server (View layer) is a thin layer of graphical user interface that passes user queries to the PolySearch2 API and formats search results. Implementing the MVC design allowed us to decouple the logic for maintaining document collections, performing searches, and presenting results to users. As a result, we can update an individual layer without affecting other layers. PolySearch2 has been tested on a variety of platforms and is compatible with most common modern browsers (FireFox, Safari, Internet Explorer, Edge, and Chrome) on both computer workstations and mobile devices (tablets and smart phones). PolySearch2's analytical algorithm was implemented in Python and it uses ElasticSearch (see Figure 15) to manage the document repository and cache results.

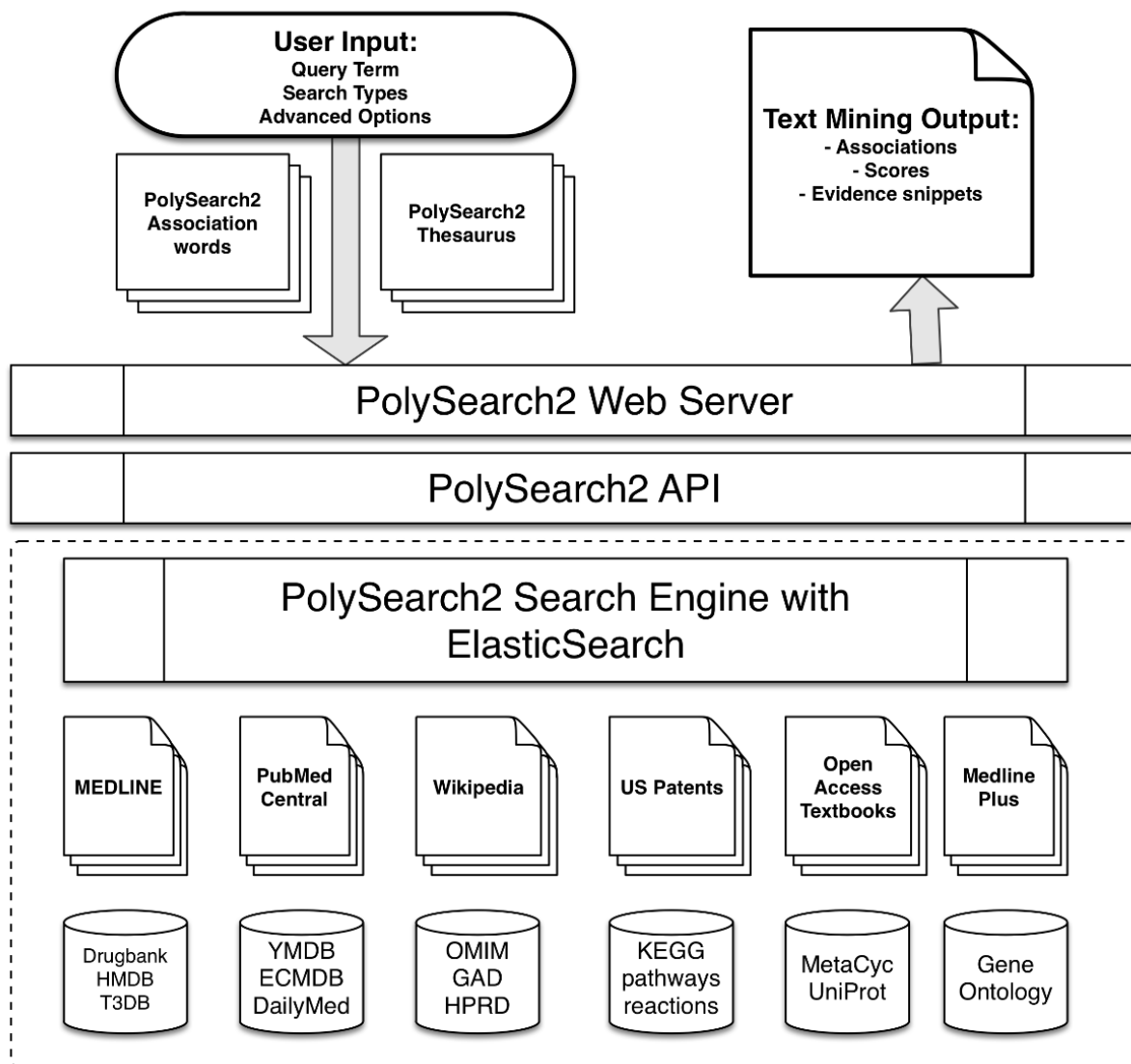


Figure 15: PolySearch2's system overview showing the architecture of the PolySearch2 web server, its API, and the underlying search engine. PolySearch2 uses the Model-View-Controller (MVC) design: 1) the PolySearch2 Search Engine with ElasticSearch (Model layer) organizes document collections. 2) the PolySearch2 API (Controller layer) implements the core PolySearch2 algorithms and queries the model layer for search results. 3) the PolySearch2 web server (View layer) is a thin layer of graphical user interface that passes user queries to the PolySearch2 API and formats search results.

3.3.3 Database and Text Search Enhancements

In PolySearch2 (<http://polysearch.ca>), we completely re-implemented the underlying text-mining framework based on the latest search engine technology (ElasticSearch, <http://www.elasticsearch.org/>) (See Figure 15). The utilization of ElasticSearch allowed us to internally host all text collections and databases (totalling 165 Gigabytes) across an ElasticSearch cluster running multiple nodes, and efficiently retrieve relevant documents. This has led to the ability to search against all thesaurus types simultaneously leading to a significant performance improvement and a nearly 25X acceleration in search speed.

In PolySearch2 we significantly expanded the number of text collections and databases (by more than 80%) to include a total of 6 free-text collections and 14 popular, text-rich bioinformatics databases. *Table 2* shows the statistics for PolySearch2's database and document collection statistics. The latest release of PolySearch2 searches against over 43 million articles covering MEDLINE abstracts, PubMed Central full-text articles, Wikipedia full-text articles, US Patent abstracts, open access textbooks from NCBI and MedlinePlus articles. We believe these free-text collections cover a wide range of human knowledge from general information (Wikipedia, textbooks and MedlinePlus), to more specific biomedical knowledge (MEDLINE and PubMed Central), to technical innovations (US Patent applications).

While free-text collections represent a body of implicit knowledge, biomedical databases represent more specific or more quantitative, high quality curated knowledge. As illustrated in the original PolySearch paper [17], incorporating relevant database records into the search greatly enhances the resulting accuracy. To further improve on the performance of PolySearch2, we incorporated DrugBank (a popular drug and drug metabolite database) [51], HMDB (a human metabolite database) [98], T3DB (a toxin and toxin-target database) [97], YMDB (a yeast metabolome database) [44], ECMDB (an E. coli metabolome database) [35], OMIM (Online Mendelian Inheritance in Man) [36], the UniProt database [92], the Human Protein Reference Database [63], DailyMed (FDA-approved drug listing information database) [66, 67], KEGG reactions and pathways [47], and the MetaCyc [15] metabolic pathway database. For more information on PolySearch2's text collections and databases sources, please consult PolySearch2's Documentation web pages.

Data Source	Database Descriptions	Number of indexed Records
OMIM	Online Mendelian Inheritance in Man	23,219
T3DB	Toxin and toxin target database	3,713
HMDB	Human Metabolome Database	41,513
MEDLINE	PubMed Abstracts	27,208,664
Wikipedia	Wikipedia abstracts	7,619,689
USPTO	US patent application abstracts	7,996,999
FooDB	Food Metabolite Database	27,509
KEGG Reactions	Kyoto Encyclopedia of Genes and Genomes	9,538
GO	Gene Ontology	40,535
DailyMed	FDA label information on marketed drugs	2,745
KEGG Pathways	Kyoto Encyclopedia of Genes and Genomes	456
NCBI Books	Full-text textbooks on NCBI bookshelf	19,066
MedlinePlus	Medical encyclopedia and dictionary	1,901
PMC	PubMed Central full-text articles	704,539
UniProtKB	UniProt Protein Knowledgebase	541,561
MetaCyc	Metabolic database for pathways, enzymes, metabolites, and reactions.	3,810
GAD	Genetic Association Database	167,298
HPRD	Human Protein Reference Database	18,863
DrugBank	Drug and drug metabolite database	6,825

Table 2: Database and Text Collection Statistics for PolySearch2. PolySearch 2.0 significantly expanded the number of text corpora and databases (by >80%) to include a total of 6 free-text corpora and 14 bioinformatics databases. The latest server searches against over 43 million articles covering Medline abstracts, PubMed Central full-text, Wikipedia articles, US Patent abstracts, and open access textbooks.

3.3.4 Improved Synonym Collections

PolySearch2's custom thesauri or synonym collections are critical for the detection of biomedical terms mentioned in its databases and text collections. The original version of PolySearch had a thesaurus that consisted of 9 categories with 57,706 terms, including names and/or synonyms for genes/proteins, gene families, diseases, drugs, metabolites, pathways, tissues, organs, and sub-cellular organelles or structures. In PolySearch2, we have significantly expanded the number of thesauri from 9 to 20 categories, and from just 57,706 terms to over 1.13 million term entries with more than 2.84 million synonyms.

PolySearch2's thesaurus collection now includes terms and synonyms for toxins [97], food metabolites [96], biological taxonomies [66, 67], Gene Ontology terms [6], MeSH terms and MeSH compounds [81], along with ICD-10 (International Classification of Disease) medical codes [8]. *Table 3* shows PolySearch2's thesaurus statistics. PolySearch2's gene/protein thesaurus and gene family thesaurus were compiled from the latest release of UniProt [25], Entrez Gene [27], the Human Genome Organisation Gene Nomenclature Committee [75], and the Human Protein Reference Database (HPRD) [63]. The disease thesaurus was compiled from the Online Mendelian Inheritance in Man (OMIM) and the Unified Medical Language System (UMLS) [42]. PolySearch2's drug and metabolite thesauri were compiled from the latest version of DrugBank [51] and the Human Metabolome Database (HMDB) [98], respectively. PolySearch2's pathway thesaurus was derived from names used for KEGG pathways [47] while PolySearch2's tissue thesaurus and organ thesaurus were created manually and the sub-cellular localization thesaurus was derived from the HPRD [63]. PolySearch2's toxin thesaurus and food metabolite thesaurus were compiled from the latest version of the Toxic Exposome Database (T3DB) [97], and FooDB (<http://foodb.ca/>) [96] respectively. The biological taxonomy thesaurus was derived from NCBI's taxonomy archive [66, 67]. PolySearch2's thesauri also feature many manually curated terms and synonyms for positive health effects, adverse health effects, drug actions, drug effects, and chemical taxonomies. All of these thesauri may be searched via PolySearch2's Thesaurus page, and all may be downloaded via PolySearch2's Download page.

Thesaurus Name	Number of Terms	Number of Synonyms
Gene Families	404	948
Adverse Health Effects	135	711
Health Effects	161	507
Gene Ontology	40,535	110,477
Toxins	3,713	39,095
Biological Taxonomy	607,031	775,728
Drugs	7,670	37,331
ICD-10 Codes	91,737	155,331
Chemical Ontology	4,017	10,098
Tissues	954	984
MeSH Terms	26,956	215,327
Food Metabolites	27,509	39,278
Genes and Proteins	27,994	287,827
Drug Effects	424	590
Metabolic Pathways	456	456
MeSH Compounds	221,986	716,676
Human Metabolites	41,793	381,195
Organs	104	201
Subcellular Locations	74	175
Diseases	27,658	76,001
Total	1,131,328	2,848,936

Table 3: PolySearch2 Thesaurus Statistics. PolySearch 2.0 significantly expanded custom thesauri from 9 to 20 categories, and from just 3000 to over 1.13 million term entries. In particular, we have expanded the thesauri to include toxins, food metabolites, biological taxonomies, pathways, as well as Gene Ontology, MeSH terms, and ICD-10 codes. The thesauri also feature many manually curated terms and synonyms for health effects, drug effects, adverse effects, and chemical taxonomies. This table summarizes the number of term entries and synonyms for each thesaurus.

3.3.5 Caching and Auto-Updating

PolySearch2 features significantly expanded support for results caching and automated updating over the original version of PolySearch. Caching allows PolySearch2 to archive the results of common queries made by users so that if the same query is made by another user, then only a trivial update (if any) needs to be performed over the previously cached material. This leads to nearly instantaneous (1-2 sec) results for many common associative queries. PolySearch2 also regularly queries itself with thesaurus terms to increase its cache coverage far beyond what users may commonly generate.

The original version of PolySearch accessed the content of all (or nearly all) of its databases via the web. This ensured absolute data currency for all its databases, but it slowed the operation down substantially as all queries were subject to problems due to heavy website traffic loads, intermittent internet outages, varying data download speeds and the extra time needed to download large data sets over the web. Because PolySearch2 searches locally maintained databases on a (very large) local disk, none of these download or web access issues are encountered. However, moving to local databases meant that the data currency problem had to be addressed. Consequently, a number of custom scripts and “Cron” jobs were developed so that new documents and new database updates are automatically retrieved on a daily basis and indexed to ensure that PolySearch2's text collections always contain the documents or data that are no more than 24 hours old.

3.4 Performance Evaluation

To assess the performance of PolySearch2, we conducted a speed test comparing only the speed of the original PolySearch with PolySearch2 on various queries with equivalent parameters. We then performed four evaluations on PolySearch and PolySearch2 to compare their accuracy. Finally, four additional evaluations were conducted to assess the performance of PolySearch2 on several novel search tasks. Performance statistics including Precision, Recall, F-measure, and Accuracy are presented in Table 4 for all 8 evaluations. Evaluation No. 1 assesses PolySearch2's ability to identify disease-gene association. Evaluation No. 2 evaluates

PolySearch2's ability to identify drug-gene/protein associations. Evaluation No. 3 assesses PolySearch2's ability to identify protein-protein interactions. Evaluation No. 4 evaluates PolySearch2's metabolite-gene associations. Evaluation No. 5 assesses PolySearch2's ability to identify drugs with significant adverse effects, or dangerous drugs". Evaluation No. 6 evaluates PolySearch2's ability to identify toxin-disease association. Evaluation No. 7 assesses PolySearch2's ability to identify toxin-adverse effect associations. Finally, Evaluation No. 8 evaluates PolySearch2's ability to identify associations to diseases given natural language question queries. All 8 evaluation datasets are available on the "Download" page on the PolySearch2 website.

We first evaluated PolySearch2's performance on four gold standard datasets (Table 4, Evaluations 1-4). Specifically, we evaluated PolySearch2's performance in mining: 1) disease-gene associations, 2) drug-gene associations, 3) protein-protein interactions, and 4) metabolite-gene associations. PolySearch2's F-measures in these tasks were 88.95, 89.75, 93.79, 90.74, respectively. Compared to the original PolySearch system, PolySearch2 achieved a 3-12% improvement in its association accuracy.

Next, we evaluated PolySearch2's performance on three new gold standard datasets (Table 4, Evaluations 5-7). These tests were designed to identify 5) adverse drug effect associations for identifying 'dangerous drugs', 6) toxin-disease associations, and 7) toxin-adverse effect associations. Performance statistics for the legacy PolySearch are not available for these datasets due to the novel search types and the size of the testing dataset. PolySearch2's F-measures on these tests were 85.85, 84.17, and 76.89 respectively.

Finally, to assess the flexibility of PolySearch2, we conducted an association test using BioASQ [35], a biomedical semantic Question Answering challenge's gold standard training dataset (Task 3B Training Set, released March 2015), and assessed PolySearch2's performance in finding associated disease concepts when presented with free-text sentences. Evaluation 8 (Table 4) shows PolySearch2's performance evaluation using the BioASQ Task 3B (biomedical semantic QA) gold standard training dataset. The search queries are question sentences from BioASQ and PolySearch2's disease association results are compared with tagged disease concepts in the BioASQ 3B gold standard training data set.

Table 5 lists some of the key feature differences between PolySearch and PolySearch2. Compared to PolySearch, PolySearch2 has a significantly expanded thesaurus (2x more categories, 19x more terms), a much larger collection of filter words (4x increase), more databases (2x increase) and many more text corpora (6x increase), as well as supporting more (4x increase) search types. We also compared both systems with regard to their analysis speed. In the speed test we calculated the speed-up factor by dividing the execution time of the old PolySearch by the execution time of PolySearch2 on an identical set of 10 search queries. Both systems were located in the same network and both were accessed over the Internet. The cache look-up was disabled on both systems. The evaluation was carried out with 10 arbitrary keywords having more than 10,000 potentially relevant documents. The keywords were "Autism, Acetaminophen, Influenza, Rheumatoid Arthritis, Escherichia coli, Vitamin, Nucleus, p53, ATP, cancer". A typical PolySearch2 query with 2,000 or fewer relevant documents was completed in less than 20 seconds. On the other hand, a typical PolySearch query was completed in 2-5 minutes. We found that the time that both PolySearch and PolySearch2 take for keywords and search types is quite consistent, so document size is actually the main factor in determining overall execution time. Based on our data, PolySearch2 achieved a 5x to 25x speedup over PolySearch, depending on the number of documents (from 500 to 10,000) it analyzed. In general, the more documents that are analyzed, the greater the speedup, as PolySearch2's initialization overhead is amortized across a larger number of document analysis. The above result shows that PolySearch2 is substantially faster, more efficient and somewhat more accurate than the original PolySearch system. The improvement in computational efficiency is primarily due to the fact that we internally host all text collections and databases in PolySearch2. In the original PolySearch, all queries were conducted through web-based APIs (which required querying and downloading abstracts from NCBI) or screen scraping on-line databases which is inherently slow. The automated update function in PolySearch2 helps ensure the currency of our document collections. The improvement in association accuracy can be attributed to the tightness measure we introduced to further discriminate matched association patterns, the assignment of weight boosting to database records in contrast to text articles, and the imposition of more stringent cut-offs to boost precision at the expense of recall (precision-recall trade-off).

Prediction Accuracy	PolySearch				PolySearch2			
	P	R	F	Acc.	P	R	F	Acc.
No. 1 Disease/Gene	0.6533	1.0000	0.7903	0.6533	0.8708	0.9091	0.8895	0.8525
No. 2 Drug/Gene	0.7490	1.0000	0.8565	0.7490	0.9701	0.8351	0.8975	0.8571
No. 3 Protein/Protein	0.8396	1.0000	0.9128	0.8396	0.9432	0.9326	0.9379	0.8962
No. 4 Metabolite/ Gene	0.7834	1.0000	0.8785	0.7834	0.9579	0.8619	0.9074	0.8614
No. 5 Drug/Adverse Effects	-	-	-	-	0.9233	0.8022	0.8585	0.7737
No. 6 Toxin/Disease	-	-	-	-	0.9054	0.7864	0.8417	0.7810
No. 7 Toxin/Adverse Effects	-	-	-	-	0.8808	0.6822	0.7689	0.7854
No. 8 BioASQ Question/ Disease	-	-	-	-	0.7284	0.6052	0.6611	0.7212

Table 4: Performance evaluation of PolySearch2 vs. PolySearch. P stands for Precision, R stands for Recall, F stands for F-measure, and Accu. Stands for accuracy.

	PolySearch	PolySearch2
Thesaurus categories	9 categories	20 categories
Thesaurus terms	57,706 terms with 353,862 synonyms	1,131,328 terms with 2,848,936 synonyms
Filter words	7011	29,718
Database Numbers	1 free-text collection and 6 databases	6 free-text collections and 14 databases
Num. of Search Types	66 query combinations	308 query combinations
Analysis Speed	6.5 documents per second	165 documents per second
Mobile Friendly?	No	Yes

Table 5: Performance evaluation and feature comparison of PolySearch2 vs. PolySearch.

3.5 Limitations

No text mining system is perfect and certainly PolySearch2 is not without some limitations. One notable limitation is its inability to progressively or interactively adapt to specific search needs. High-end search engines such as Google and Yahoo monitor user-feedback through surreptitious monitoring of user mouse clicks, web-page access and web-page dwell times. This helps these search engines customize or adapt to user preferences and needs. Ideally PolySearch2 should be able to adapt to a search task by considering user feedback on the quality of discovered associations. For example, users may indicate certain associations to be false positives and in subsequent runs PolySearch2 should ideally learn from these negative examples and adapt itself to match a user's specific search needs and thereby achieve higher accuracy. We are currently testing several feedback systems and considering adding a “search satisfaction” feedback system in future versions of PolySearch2. Another limitation with PolySearch2 (and for most text mining systems) is its inability to self-assess its results and to extract specific knowledge on its own. While PolySearch2 performs well at extracting strong

associations between biomedical entities it is not yet capable of assessing its discovered associations or extracted relations. For example, PolySearch2 is able to identify a potential association between BPA and breast cancer but it is not able to infer a cause-and-effect relationship from the discovered association. Part of this limitation is due to the lack of training data to perform assessments and to extract relationships. To address this issue, we are hoping to use Machine learning (ML) and Natural Language Processing (NLP) techniques to eventually convert PolySearch2 from a simple association discovery tool to a more general knowledge extraction tool. We are currently working to incorporate this capability into future releases of PolySearch2.

3.6 Conclusion

In this chapter we have described PolySearch2 (<http://polysearch.ca>), a web server designed to facilitate data mining and the semi-automated discovery of text associations between a wide range of biomedical entities. PolySearch2 supports “Given X, find all associated Ys” type of queries with X and Y from more than 20 types of biomedical subject areas including human diseases, genes, SNPs, proteins, drugs, metabolites, toxins, metabolic pathways, organs, tissues, subcellular organelles, positive health effects, negative health effects, drug actions, Gene Ontology terms, MeSH terms, ICD-10 medical codes, biological taxonomies and chemical taxonomies. Some of the most significant improvements for PolySearch2 include a significant modernization of its underlying text-mining framework; a complete upgrade and re-implementation of the web interface using the latest web technology standards; a substantially improved algorithm for improved scoring and ranking of associations; significantly expanded custom thesauri and term collections; an expanded number of text collections and databases (by >80%); along with significantly improved support for caching and automated updating. PolySearch2 now offers greater speed (up to 25X faster), accuracy (3-12% improvement on f-measures), customizability (additional configurable options) and usability (modern and mobile-friendly web interface) than the original version. We believe that with these recent enhancements, PolySearch2 can better facilitate text-based discovery (and re-discovery) of latent associations among many types of biomedical entities and topics.

4. BioQA: An Automated Biomedical Question Answering System

Biomedical information is growing rapidly thanks to steady advances in both biological and medical technologies. Most biomedical information is archived in the form of free-text in peer-reviewed publications, or stored in various electronic databases using a variety of different text-based formats. Our ability to find relevant biomedical records or articles has been greatly accelerated by the development of specialized biomedical search engines like NCBI Entrez or Google Scholar. However, in order to keep pace with a specific biomedical field or to find answers about specific biomedical questions, researchers still need to construct large numbers of Boolean queries using a special lexicon of appropriate key words and then manually scan through dozens of irrelevant articles just to find the one pertinent paper or the one key finding. This is very inefficient. What is needed is a “wise” biomedical question-answering system to assist researchers in finding relevant articles or answering specific biomedical questions. Such a system would eliminate the time consuming task of manual scanning and make the challenges of finding relevant information or answering specific questions far more efficient. In this paper, we introduce BioQA, a biomedical question-answering system, as an initial solution to the biomedical question-answering task. The BioQA framework specifically organizes biomedical information for fast and precise retrieval, and comprises of various algorithms to transform natural language questions into natural language answers. BioQA is capable of processing natural language questions, performing searches across both free-text collections and various biomedical databases, and automatically summarizing the answers with supporting evidence. We specifically developed BioQA to handle both descriptive and associative queries. The BioQA web server is publicly available online at <http://bioqa.ca>.

4.1 Introduction

Biomedical information is growing at an explosive rate. As a result, it is increasingly difficult for researchers to keep pace with this rapidly growing body of information [9]. For example, PubMed, which contains more than 25 million indexed abstracts from more than 5,140 journal titles, is growing at rate of 4% each year, and more than 3,000 new articles each day [54].

GenBank [18], which contains most of the world's gene sequencing information, has grown from just 600 annotated DNA sequences in 1982 to nearly 200 million annotated DNA sequences today. The Protein Data Bank [69], which houses most of the world's protein structure data, grew from 13 structures in 1976 to more than 120,000 structures by 2015. ArrayExpress [70], which contains data on gene expression experiments, grew from just 1,200 data sets in 2006 to nearly 70,000 today. Adding to the challenge of exponential information growth, is the proliferation of domain-specific databases. For instance, the total number of biomolecular databases ever described in the annual Nucleic Acids Research (NAR) Database Issue has grown from 90 in 1998 to nearly 1700 today [80]. Each database uses its own schema and therefore each resource needs to be accessed or searched according to its own specific query system. To address these growing problems of database proliferation and database size, a number of groups have started to develop aggregative biomedical search engines or smarter text mining tools. These include such systems as NCBI Entrez [66, 67], GoPubMed [22], and PolySearch2 [16, 17]. However, even with these powerful software tools, researchers still need to manually scan through (potentially hundreds of) articles and database records to find answers to simple questions, or to find the supporting evidence needed to advance an idea. This bottleneck of manual text scanning has arisen because most biological knowledge, whether it is in papers or in databases, is buried in the form of free text. This means that queries or questions must be constructed as primitive Boolean word queries or Boolean word combinations. The results are typically lists of records with differing levels of text matches and widely varying levels of relevance.

Ideally what is needed to overcome this “free text bottleneck” are software tools that can efficiently mine biomedical data and rapidly extract or compose answers from relevant snippets of information. One approach involves the development of a “wise” biomedical question-answering (QA) system. Such QA system would ideally accept free text questions and provide precise free text answers with encyclopedia-like commentary and appropriate references or attribution. Research on developing computer-based QA systems has become increasingly popular in recent years, following the success of Watson, an IBM-developed QA system [28]. Watson came to prominence by defeating highly skilled human players on the open-domain question answering *Jeopardy!* challenge. The success of Watson has motivated many text mining

experts to start developing question answering systems tailored for other applications beyond general knowledge or game show trivia. One particular area of interest has been the development of QA systems for enhancing biomedical research. BioASQ [82] is a biomedical semantic indexing and question answering challenge aimed at accelerating the field of biomedical question answering through competitive shared tasks. Two shared tasks are available: 1) indexing novel MEDLINE abstracts using MeSH terms (Task A), and 2) retrieving concepts and snippets to form natural language answers (Task B). A number of systems have been developed for the BioASQ challenges including the BioASQ baseline system, the MCTeam system, a modified NCBI system and BioQA (described here). The BioASQ team developed a “baseline” system to compare with participating teams [88, 89]. The baseline system retrieves the top 50 and top 100 concepts and snippets returned from their search system, formulates a final answer using greedy and Integer Linear Programming algorithms, and further selects candidate answers using Support Vector Regression [82, 88]. The MCTeam system [103] participated in the BioASQ challenge and this system used MetaMap [5] to identify concept-related words in input query and formulated a search query to query a local index of PubMed full-text articles and merge retrieved results to final answers. The NCBI system [57] used the PubMed search function to retrieve relevant documents and snippets from MEDLINE abstracts, and a dictionary look-up method to recognize concepts and resolve concepts to MeSH / Gene Ontology terms using GenNorm [94] and MetaMap [5]. The NCBI system then used the PubTator tool to generate and rank candidate answers [93, 95]. These tools have been tested and compared through several shared-task biomedical QA challenges like BioASQ [82, 88, 89]. Competitions such as BioASQ have certainly helped to advance the field of biomedical information retrieval and question answering. However, biomedical QA is still facing two core challenges: 1) biomedical information is stored in widely dispersed databases in highly heterogeneous formats that make information searching and consolidation difficult, and 2) a significant portion of biomedical information is represented in the form of free-text, which needs extensive text processing to extract useful information.

In taking on both challenges, we have developed BioQA. BioQA is a biomedical question answering system, capable of handling natural language queries and providing comprehensive natural language answers with supporting evidence. In particular, BioQA is able to handle descriptive (“*What is Aspirin?*”) and associative (“*What is the cause of beri-beri?*”) queries.

Descriptive queries are particularly useful for biocurators needing assistance in annotating genes, proteins, metabolites, and other biomedical entities, while associative queries are useful for finding latent associations between biomedical entities. BioQA is able to automatically summarize relevant documents and passages and, in doing so, it is also able to generate supporting evidence for the returned answers to assist researchers in analyzing the extracted results. We specifically designed BioQA to focus on answering biomedical questions posed by researchers, medical practitioners, students, and the inquisitive public. BioQA is available to the public on <http://bioqa.ca>.

In this chapter, we described the BioQA public web interface and its underlying question answering framework. We also discuss, in detail, how BioQA manages and organizes heterogeneous biomedical knowledge, as well as the system of algorithms enabling BioQA to process natural language queries and relevant text passages. This includes a discussion of how BioQA identifies biomedical Named Entities (NEs), how it analyzes free text questions to form search queries, how it retrieves relevant documents and databases records, how it synthesizes descriptions and how it summarizes and paraphrases natural language answers. Finally, we evaluate various components of BioQA with the BioASQ challenge datasets and discuss BioQA's limitations and future directions.

4.2 BioQA's User Interface

BioQA (<http://bioqa.ca>) features a graphical web interface designed to work on both computer workstations and mobile devices. Figure 16 to Figure 22 show various pages from BioQA's web interface. Figure 16 shows the question submission page where a user can post a question to BioQA to search for relevant concepts and retrieve BioQA's answers. Users can post a question to get a "Quick Answer" or a "Full Answer" from BioQA. A "Quick Answer" query searches document collections and BioQA's knowledge base for relevant concepts, descriptions, and information snippets, while a "Full Answer" query performs an additional query to PolySearch2 to obtain more relevant concepts and information snippets. Upon submitting a question, a user will be redirected to an auto-refreshing query processing page while the question is being analyzed and answers are being acquired in the background. Depending on the query

type and the number of documents that need to be analyzed, a specific query can take from 30 seconds to a few minutes to process. The BioQA web server caches its results for 7 days and users can use the assigned search id to look up and retrieve the cached results using the “Check Result” page (not shown in this figure). Once a query has been completed, the user will be redirected to an “Answer Synopsis Page” as shown in Figure 17. The synopsis page is a hub with links to full textual answers (Figure 18 and Figure 19), relevant concepts (Figure 20), and knowledge graphs (Figure 21 and Figure 22). The synopsis page features a tag cloud generated via frequently used words from the retrieved snippets to provide a quick visual graphic of the text answers. The font size of the words in the tag cloud are proportional to frequencies of occurrence in the relevant text snippets. The synopsis page also shows the original questions and a short preview of the full answer. A navigation bar with light grey background (Figure 19) is provided for users to quickly review and navigate within the result hierarchy. BioQA results are also available in JSON format for download. These features are described in more details on BioQA’s Documentation web page.

Clicking on the link marked “Full Answer with References” takes the user to the “Answers with References” page showing the full textual answers to user’s posted question. Textual answers are formatted into different paragraphs providing information on entity descriptions as well as their relationships to each other in the context of the posted question. The answer text is color-coded (according the type of recognized biomedical entities) and hyperlinked (to relevant external biomedical databases). Moreover, sentences in the answer text are annotated with references to the original documents in BioQA’s text collections, and these references are hyperlinked to original articles in the corresponding databases (including PubMed, PMC, or Wikipedia). For example, Figure 18 shows full BioQA answers to the question “*What is the cause of beri-beri?*”. The first paragraph in this answer defines beri-beri as a cluster of symptoms and it is caused by Vitamin B1 (thiamine) deficiency. The BioQA answer also lists other related diseases caused by thiamine deficiency, and provides some history on how the association between beri-beri and thiamine deficiency was discovered. Figure 19 shows full BioQA answers to the question “*What diseases are caused by E-cadherin mutations?*”. Similarly, this answer first defines E-cadherin as a “calcium-dependent cell adhesion molecule”, describes its molecular function and its association with breast cancer. Checking the relevant diseases

concepts provided by BioQA (Figure 20) we can see that E-cadherin is found to be associated with “malignant tumoral disease”, “gastric cancer”, “breast neoplasm”, “adenocarcinoma”, “melanoma”, “prostate neoplasms”, and other cancers.

BioQA features an automated paraphrasing function (Figure 18) to automatically paraphrase sentences (derived from previously published or copyrighted works) in text answers. This may be used by users wishing to avoid copyright/plagiarism issues and to help them better integrate text snippets into their own work. Clicking on the “Paraphrase Answer Text” link will initialize a paraphrasing operation on the initial BioQA answers and, upon completion, the paraphrased answer will be displayed again on the same “Answers with References” page with the original references. It is worth noting that this automated paraphrasing operation randomizes paraphrasing results, so clicking on “Paraphrase Answer Text” again will generate a different set of paraphrasing results. (Please refer to Chapter 5 for algorithmic and implementation details on BioQA’s paraphrasing function.) BioQA also supplements the generated textual answer with list of relevant concepts and a concept network graph for visualization. Clicking on the “View Relevant Concepts” (Figure 17) button takes user to the “Relevant Concepts” page (Figure 20) which shows the associated entities retrieved by PolySearch2 [52]. The full list of relevant categories includes human diseases, genes, SNPs, proteins, drugs, metabolites, toxins, metabolic pathways, organs, tissues, subcellular organelles, positive health effects, negative health effects, drug actions, Gene Ontology terms, MeSH terms, ICD-10 medical codes, biological taxonomies, and chemical taxonomies. Relevant concepts in this page are organized by their categories and clicking on each category tab (marked with number of found relevant entities) displays a relevant concept table for that particular category. Each relevant concepts table is sorted by the Z-scores in descending order, and each list can be sorted by clicking on the column header. The relevant concept table lists the Z-score and PolySearch2 Relevancy score (R-score) as well as the name, synonyms, and number of supporting text snippets for each associated entity. Each entry is hyperlinked to external database records. Users can also download the relevant concepts results in JSON format to get further details on the supporting text snippets (hits).

In addition to textual answers and tables for relevant concepts, BioQA also provides another form of answer representation: concept graphs. Concept graphs allow one to easily visualize relevant biomedical concepts and their relationships. These can include relationships

such as being co-mentioned in relevant text snippets (a co-mentioned graph), or relationships derived by being referenced across biomedical databases (a knowledge graph). Figure 21 and Figure 22 show screenshots of BioQA's concept network graphs. On the Answer Synopsis Page (Figure 17), users can choose the format for the graph layout (grid, circle, breadth-first, etc.) from the drop-down list, and click "Visualize co-mention graph" or "Visualize knowledge graph" to view and interact with a concept graph image. In any concept graph layout (Figure 21), users can use their mouse or track pad to zoom in/out, re-position the whole graph, reorganize individual nodes to achieve better viewing angles or further inspect interesting clusters of nodes or edges. Figure 22 shows a zoomed-in view of an example concept network component. Nodes in a concept graph represents a biomedical concept or entity. Nodes are color-coded based on concept or entity types and are hyper-linked to corresponding database records. Edges in the concept graph represent relationships among concepts. These relationships may include being frequently co-mentioned in sentences among relevant text snippets in co-mentioned graphs, or being cross-referenced in annotation entries across different biomedical databases. By reviewing the relationships among the concepts in a concept graph, users may discover hidden relationships between two biomedical entities that are connected via some other biomedical entities. Such long-range relationships may not be easily detected using text-mining analysis (e.g. PolySearch2), which tend to focus on entities co-mentioned within a sentence. Users can review concepts in concept graphs in concept tables by clicking on "View Co-mentioned Graph Nodes" or "View Knowledge Graph Nodes" buttons. A concept table shows concept types, the concept ID, the concept name (hyperlinked), and synonyms for concepts represented on the corresponding concept graphs. Both co-mentioned graphs and knowledge graphs are available for download in JSON format. Under the hood, BioQA uses generated concept graphs to discover relationships between entities for the posted question and to generate appropriate textual answers. For more information on how BioQA generates its concept graphs and how BioQA's graph-based summarization algorithms work, please refer to the Algorithms section in this paper or BioQA's documentation page. Also see Chapter 5 for implementation details.

BioQA [Ask BioQA](#) [Check Result](#) [Documentation](#) [Downloads](#) [About](#) [Contact Us](#)

Welcome to BioQA

BioQA is a biomedical question answering system, capable of handling natural language queries and providing comprehensive natural language answers with supporting evidence. In particular, BioQA is able to handle descriptive and associative queries. BioQA is able to automatically summarize relevant documents and passages and, in doing so, it is also able to generate supporting evidence for the returned answers to assist researchers in analyzing the extracted results. We specifically designed BioQA to focus on answering biomedical questions posed by researchers, medical practitioners, students, and the inquisitive public.

[Get Started](#)

Tweets by @WishartLab

Wishart Lab @WishartLab
Retweeted Daniel Himmelstein (@dhimmel):
Kudos to @OMxInc & @cknoxrun for fixing the #DrugBank licensing. The...
<fb.me/2c91K4Upp>

22 Jun

Wishart Lab Retweeted

[Embed](#) [View on Twitter](#)

Ask BioQA

To use this server:

1. Simply enter your question
2. Press "Quick Answer" to start a search, OR
3. Press "Full Answer" to start a search including [PolySearch2](#) results

Given Question [Quick Answer](#) [Full Answer](#)

Please cite:

Liu, Y., Wilson, M., Liang, Y., Djoumbou, Y., Arndt, D., Sajed, T., Wishart, D.S. (2016) BioQA: a web-based automated biomedical question answering system (manuscript in preparation).

Figure 16: BioQA’s Query submission page (the Question is: “What is the cause of beri-beri?”).

Question

What is the cause of beri-beri

Answers

Beriberi refers to a cluster of symptoms caused primarily by a nutritional deficit in **Vitamin B1** (thiamine). Beriberi has conventionally been divided into three separate entities, relating to the body system involved (nervous or cardiovascular) or age of patient (infantile). Beriberi is one of several thiamine-deficiency related conditions, which may occur concurrently, including Wernicke 's encephalopathy, Korsakoff 's syndrome, and Wernicke-Korsakoff syndrome. [1]

Tropical ulcers -- which are often diphtheria appearing as a secondary infection of a skin disease -- were a common medical complaint, along with dysentery, malaria, beri-beri, dengue, scabies, and septic bites and sores. [1] beri weakness, the reduplication being intensive ..., page 203, 1937 A Sinhalese-English Dictionary, Rev. There were also the presence of cholera, influenza, smallpox, beri-beri, dysentery, bubonic plague, scurvy, rheumatism, asthma, syphilis, tetanus, toothache, and ulcers. [1] A dramatic example of the effect of food processing on a population 's health is the history of epidemics of beri-beri in people subsisting on polished rice. [3] Removing the outer layer of rice by polishing it removes with it the essential vitamin thiamine, causing beri-beri. [3] McCarrison 's work on goitre, cretinism, and the thyroid, begun in the western Himalayas in 1902, generated scores of scientific publications during the following thirty-five years, While McCarrison 's work is often considered the start of serious studies of goitre and cretinism in South Asia, it was preceded by that of Commissioner David Scott at Rangur in north-east India around 1825, and was investigated by Mountford Bramley at Kathmandu in 1832. [5] In 1918, McCarrison founded the Beri-Beri Enquiry Unit in a single room laboratory at the Pasteur Institute in Conoor, India. [5] : Rice is 370 kcal/100 grams, and the average person needs something like 2000 kcal/day. [7] Assuming that all of the various micronutrients could be provided in pill form (so we could avoid scurvy and beri-beri and that stuff) a person would need $2000/370 \times 100 = 540$ grams of dry rice. [7]

He showed that these values, together with the incidence of beri-beri, justified the application to man of a formula obtained experimentally and gave an indication of man 's minimum requirement of the vitamin. In this paper the vitamin B (1) value of the same diets has been obtained from direct assays against the International Standard of the constituent foods, raw or cooked, as usually eaten. [1] Twelve women groups in 10 villages of block Beri were identified and activated through participatory health communication actions for mother and child development. [2] Finally, nutritional disturbances and metabolic diseases, such as Kwashiorkor, beri-beri, obesity, alcohol consumption, and diabetes mellitus may also lead to severe cardiac dysfunction. [3] outline the importance of early diagnosis and treatment for this form of high-output heart failure, which has a poor prognosis and, if left untreated, can determine the death of the patient in a few days. [4] Synthetic chemistry, together with improvements in the diet and in education, largely overcame scurvy, beri-beri, and pellagra, but deficiencies of vitamins A, C, and folic acid still occur widely in economically disadvantaged populations, and this is a challenge to those who wish to improve public health. [5] Beri may refer to: Beri is a town and a municipal committee in Jhajjar district in the state of Haryana in northern India. Beri is a village in Piparali tehsil in Sikar district of Rajasthan state in India. [7] A list of films produced by the Bollywood film industry based in Mumbai in 2005: Beri 10 P.N. This is a list of Major District Roads in Himachal Pradesh, India. [8] The technique was largely abandoned in later 20th century as pipes lines or hand pumps were laid, it was when faced with drought like situations, inadequate supplies of piped water on the account of growing population, which also resulted in depleted or contaminated ground water, this traditional method was revived, along with other traditional rainwater harvesting structures like, Naadi, a village pond and Beri, a small rainwater-collecting wells, especially for supplying drinking water. [9]



Paraphrase Answer Text

References

- [1] wikipedia entry Beriberi: Beriberi
- [2] wikipedia entry Batu_Lintang_camp: Batu Lintang camp
- [3] wikipedia entry Nutrition: Nutrition
- [4] wikipedia entry Robert_McCarrison: Robert McCarrison
- [5] wikipedia entry Wikipedia:Reference_desk/Archives/Science/2009_September_11: Wikipedia:Reference desk/Archives/Science/2009 September 11

Figure 18: BioQA 's full answer page (the Question is: "What is the cause of beri-beri?").



Question

What diseases are caused by e-cadherin mutations?

Answers

Cell adhesion protein **E-cadherin** is indispensable for a robust pluripotent phenotype. [1] Torben Redmer, Sebastian Diecke, Tamara Grigoryan, Angel Quiroga-Negreira, Walter Birchmeier, Daniel Besser (2011) **E-cadherin** is crucial for **embryonic stem cell** pluripotency and can replace **OCT4** during somatic cell reprogramming. [1] EMBO reports, 12, 720 - 726, doi:10.1038/embor.2011.88 During reprogramming for **iPS cell** generation, **N-cadherin** can replace function of **E-cadherin**. [1] The effect of restricted cell spreading on mESC self-renewal is not mediated by increased intercellular adhesion, as evidenced by the observations that inhibition of mESC adhesion using a function blocking anti **E-cadherin** antibody or siRNA do not promote differentiation. [1] Relative frequency of loss of **E-cadherin** and **CD44** has also been observed. [5]

E-cadherin is a calcium-dependent **cell adhesion** molecule which is important in cell-cell interactions in **epithelium** and plays a major role in maintaining the structure and integrity of **epithelial** sheets. [1] **E-cadherin** and its associated **cytoplasmic proteins** including alpha-, beta-, and **gamma-catenin** play a pivotal role in the maintenance of normal tissue architecture and the suppression of cancer invasion. [2] The reduction or loss of **E-cadherin** (E-cad), a calcium-dependent **epithelial cell adhesion molecule**, has been associated with tumor **dedifferentiation** and invasiveness. [3] Homophilic interactions of **E-cadherins** are responsible for **cell-cell adhesion** in the **adherens junctions** of the biological barriers (i.e. Here we consider the effects on E-cad expression of eight potential regulatory factors: E-cad promoter **DNA methylation**, the transcript levels of six transcriptional repressors (**SNAI1**, **SNAI2**, **TCF3**, **TCF8**, **TWIST1**, and **ZFHX1B**), and E-cad DNA copy number. [4] **E-cadherin** is a transmembrane **calcium-dependent cell-cell adhesion** molecule that is complexed with **cytoplasmic proteins** including **alpha-catenin**, **beta-catenin**, **plakoglobin (gamma-catenin)**, and **actin**. [5] The initial step of cancer invasion and metastasis is the escape of tumour cells from the primary site, involving disruption of normal **cell-cell adhesion** and **E-cadherin** (E-cad) and **beta-catenin** (beta-cat) **down-regulation**, as shown in various types of human malignancies including breast carcinomas. [6] We showed that the YMB-1-derived breast cancer **cell line** YMB-S, which proliferates in suspension without aggregation, exhibits complete loss of **cell-cell adhesion** despite the presence of E-cadherin-catenin complex and expression of free **beta-catenin** in the cytoplasm. [7] Alterations in junction-protein **phosphorylation** showed drastic loss of **E-cadherin** and **beta-catenin** in cell-cell contacts and the increase of cytoplasm and nuclear **beta-catenin** in epidermis, suggesting the activation of the **beta-catenin signal pathway**. [8] These results suggest that **vinculin** plays a role in the establishment or regulation of the cadherin-based **cell adhesion** complex by direct interaction with **beta-catenin**. [9]

Figure 19: BioQA's full answer page (the Question is: "What diseases are caused by E-cadherin mutations?").

Success! Found 138 hits with e-cadherin on 2016-09-14 15:46:01 .

Question: What diseases are caused by E-cadherin mutations?

[Ask BioQA - Search 1473889523](#) / Question: [What diseases are caused by E-cadherin mutations?](#) / [Relevant Concepts](#)

- [3 Adverse Effects](#)
- [22 Diseases](#)
- [6 Drugs](#)
- [1 Drug Effects](#)
- [1 Food Metabolites](#)
- [17 Genes/Proteins](#)
- [9 Gene Ontology Terms](#)
- [1 Health Effects](#)
- [2 Human Metabolites](#)
- [40 MeSH Terms](#)
- [6 MeSH Compounds](#)
- [6 Organs](#)
- [2 Pathways](#)
- [3 Species](#)
- [5 Subcell Locations](#)
- [8 Tissues](#)
- [4 Toxins](#)

ZScore	RScore	Entity ID	Name	Synonyms	Hits
24.46	1505	DID55595	malignant tumoral disease	malignant tumoral disease; Malignant neoplastic disease; Cancer morphology; Malignant tumor morphology; malignant tumor; Malignant Neoplasms; neoplasm/cancer; Malignant neoplasm; Malignant tumour; Tum... (Read More)	87
20.97	1300	DID02709	gastric cancer	gastric cancer; Stomach Cancers; Gastric cancer; Gastric Cancers; Malignant Neoplasm of the Stomach; Malignant neoplasm of stomach; Malignant Gastric Neoplasm; stomach cancer; Cancer of Stomach; Malig... (Read More)	60
17.64	1105	DID06582	breast neoplasm	breast neoplasm; Breast tumour; Neoplasm of breast; Breast Neoplasm; Tumors, Breast; Breast Tumor; Tumor of the Breast; Tumor, Breast; mammary tumor; Breast Neoplasms; Tumor of breast; Tumour of breas... (Read More)	54
9.71	640	DID68287	Adenocarcinoma	Adenocarcinoma; Malignant Adenomas; ADENOCARCINOMA NOS; Malignant Adenoma; Adenocarcinomas; Adenocarcinoma nos (morphologic abnormality); Adenocarcinoma, no subtype (morphologic abnormality)... (Read More)	40
6.13	430	DID08019	Melanoma	Melanoma; Melanoepithelioma; Melanoblastoma... (Read More)	20
5.36	385	DID49267	Prostate Neoplasms	Prostate Neoplasms; Tumor of the Prostate; Neoplasm of the Prostate; prostate neoplasm; Tumor of Prostate; Prostatic Neoplasms; Tumour of prostate; prostatic neoplasm; Prostate Tumor; PROSTATE NEOPLAS... (Read More)	17
4.26	320	DID34157	Squamous Carcinoma	Squamous Carcinoma; epidermoid carcinoma; Carcinomas, Squamous Cell; Carcinomas, Squamous; Squamous Cell Epithelioma; CARCINOMA EPIDERMOID; Carcinoma, Epidermoid; Squamous Cell Carcinomas; Carcinoma, ... (Read More)	20

Figure 20: BioQA’s relevant concept view for the input question “What diseases are caused by E-cadherin mutations?”.

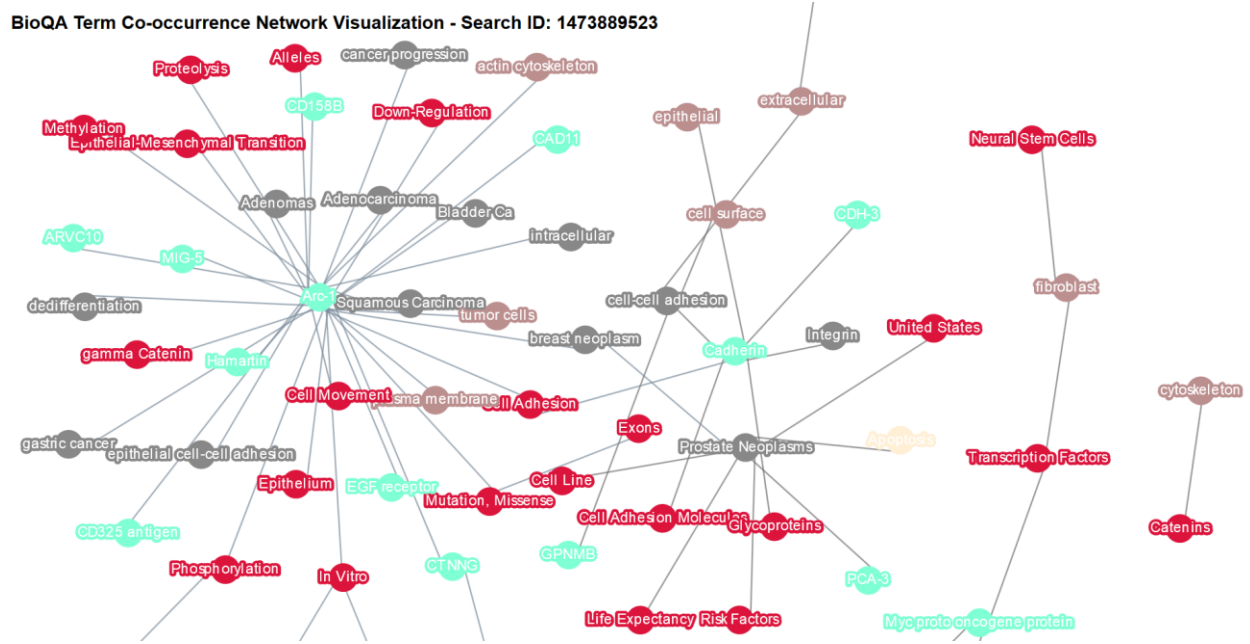


Figure 22: A close-up view on BioQA’s Co-occurrence network visualization. (The question is: “What diseases are caused by E-cadherin mutations?”)

Besides providing a public web interface to serve the general public, BioQA also offers certain underlying datasets as general resources for the biomedical community. These can be used by individuals to build their own text-mining or question answering systems. These datasets are fundamental building blocks to BioQA and we believe that by releasing them to the public it will help advance the field of biomedical question answering. In particular, BioQA uses PolySearch2’s thesauri for its entity recognition module. The PolySearch2 thesauri contain 1.13+ million biomedical terms with 2.85+ million synonyms. The complete set of terms are available for download on the PolySearch2 web server (<http://polysearch.ca>). BioQA also uses an in-house Medline N-gram dataset for question analysis. BioQA’s Medline N-gram dataset calculates the observed frequency of unigrams, bigrams (two consecutive words), trigrams (three words), 4-grams, and 5-grams from 23+ million Medline titles and abstracts (Medline 2015 Baseline + Update). Each N-gram dataset contains 7 to 74 million N-gram entries. BioQA’s Medline N-gram dataset offers finer granularity than the NLM Lexical System Group’s Medline N-gram

dataset [53]. This is because BioQA's N-gram dataset is created without character length restriction and it only filters out singletons (terms that occur only once across the entire dataset). Finally, the graphical structure (nodes and edges) in BioQA's knowledge base (BioKB), which was built by aggregating entries and cross-references across 20+ biomedical databases, is also available for download in YAML format. All aforementioned datasets, along with BioQA's evaluation datasets, are available for download on the web server's "Download" pages.

4.3 BioQA's Knowledge base

Key to BioQA's operations and success are its knowledge base and algorithmic components. BioQA encapsulates "knowledge" in various representations through a knowledge base called BioKB which consists of numerous biomedical databases, text collections, knowledge graphs and thesauri. In particular, BioKB consists of three components: 1) a comprehensive collection of biomedical thesauri, 2) a large collection of free-text documents, 3) an interconnected knowledge graph capturing relationships between biomedical concepts annotated with concept description and attributes.

Thesaurus Name	No. of Terms	No. of Synonyms	Data Sources
Genes and Proteins	27994	287,827	UniProt, Entrez Gene, HGNC, HPRD, JoChem
Gene Families	404	948	UniProt, Entrez Gene, HGNC, HPRD
Diseases	27,658	76,001	OMIM, UMLS, SNOMED CT
Drugs	7670	37,331	DrugBank
Human Metabolites	41,793	381,195	HMDB
Metabolic Pathways	456	456	KEGG
Tissues	954	984	Manual curation
Organs	104	201	Manual curation
Subcellular structures	74	175	HPRD
Toxins	3,713	39,095	T3DB
Food Metabolites	27509	39,278	FooDB
Biological Taxonomy	607,031	775,728	NCBI Taxonomy and Integr8
Gene Ontology	40,535	110,477	Gene Ontology
MeSH terms	26956	215,327	MeSH
MeSH Compounds	221,986	716,676	MeSH
ICD-10 Medical Codes	91,737	155,331	ICD-10 Codes
Positive Health Effects	161	507	Manual curation
Adverse Health Effects	135	711	Manual curation
Drug Effects	424	590	Manual curation
Chemical Ontology	4,017	10,098	Manual curation
Total	1,131,328	2,848,936	All

Table 6: Statistics for BioKBs biomedical thesauri collections. This table shows the name of the individual thesaurus, number of terms and synonyms, as well as the primary source. BioKB's thesauri includes terms and synonyms for 20 different types of biomedical entities, including genes, proteins, protein families, diseases, human metabolites, drugs and drug metabolites, biological pathways, tissues, organs, sub-cellular organelles, toxins, food constituents, biological taxonomies, ICD-10 medical codes, positive and adverse health effects, drug effects, and chemical taxonomies.

BioKB's biomedical thesauri are the foundation of other high level functionalities like biomedical term recognition, term tagging, sentence weighting, and summarization. BioKB contains a collection of 20 comprehensive biomedical thesauri with over 1.13 million terms and 2.84 million synonyms. BioKB's thesauri collection include terms and synonyms for genes, proteins [92], gene families [92], diseases [36], human metabolites [96], drug and drug metabolite [51], biological pathways [47], tissues [52], organs [52], sub-cellular organelles or structures [52], toxins [97], food metabolites [96], biological taxonomies [76], Gene Ontology terms [6], MeSH terms and MeSH compounds [81], ICD-10 (International Classification of Disease) medical codes [8], and SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) [61], as well as positive health effects [52], adverse health effects [52], drug actions [52], drug effects [52] and chemical taxonomies. All of these thesaurus terms have been checked and curated manually by ourselves and others. BioKB's thesauri overlap significantly with PolySearch2's thesauri but they also include many important enhancements. Table 6 summarizes each of BioKB's thesauri and the number of terms and synonyms in each thesaurus. BioKB's gene, protein, and gene family thesauri were compiled from the latest release of UniProt [92], Entrez Gene [18], the Human Genome Organisation Gene Nomenclature Committee (HGNC) [75], and the Human Protein Reference Database (HPRD) [63]. Furthermore, BioKB's thesauri also incorporate dictionary terms and synonyms curated by the Joint Chemical Dictionary (JOICHEM) [39] to further improve BioQA's term recognition capability. The disease thesaurus was compiled from Online Mendelian Inheritance in Man (OMIM) [36], the Unified Medical Language System (UMLS) [14], and SNOMED CT [61]. The drug and metabolite thesauri were compiled from the latest version of DrugBank [51] and HMDB [96], respectively. The biological pathway thesaurus was derived from names used for KEGG pathways [47]. The tissue thesaurus and organ thesaurus were created manually and the sub-cellular localization thesaurus was derived from the HPRD [63]. BioKB's toxin thesaurus and food metabolite thesaurus were compiled from the latest version of the Toxic Exposome Database (T3DB) [97], and FooDB [96] respectively. The biological taxonomy thesaurus was derived from NCBI's taxonomy archive and the Integr8 database [76]. BioKB's thesauri also feature many manually curated terms and synonyms for positive health effects, adverse health effects, drug actions, drug effects and chemical taxonomies.

BioKB also contains a large collection of free-text documents. This free-text document collection is the source of all text snippets for BioQA's document retrieval process. To enhance BioQA's processing speed, all of BioKB's free-text document collections are hosted internally and consist of more than 43 million free-text documents (totaling 65 Gigabytes in storage size). This avoids delays caused by posting queries over the internet to external databases. BioQA extracts relevant documents and snippets from this document collection to support downstream query processing and answer synthesis. BioQA accesses all of BioKB's free-text collections using an ElasticSearch cluster running multiple nodes [3]. Apache ElasticSearch is an open source information retrieval system that allows BioQA to efficiently retrieve relevant documents from BioKB's text collection and databases. Prior to being added to the text collection, each free-text document is analyzed, parsed, and indexed using BioKB's thesauri for rapid search and retrieval. The incorporation of the latest search engine technologies enables BioQA to search the entire BioKB document collection rapidly and to find documents relevant to a BioQA search query in just a few seconds. BioKB's document collections covers a wide spectrum of human knowledge in the form of free-text articles, ranging from general knowledge text to biomedical specific text. This document collection includes latest release of the MEDLINE abstracts, PubMed Central full-text articles, Wikipedia full-text articles, US Patent abstracts, open access textbooks from NCBI and MedlinePlus articles. Table 7 lists each free-text document collection, along with number of records and storage size requirement. The MEDLINE abstract document collection is updated automatically to retrieve latest MEDLINE abstracts online to ensure BioKB's MEDLINE abstracts collection stays current. Other document collections are updated when new releases become available. Table 8 lists each structured database, along with number of records indexed in BioKB. The cross reference portion of these databases are used to populate concept connections in the knowledge graph, while the free text portion of these database are indexed in ElasticSearch for mining concept associations.

Free-text collection	Number of Records	Storage Size
MEDLINE (PubMed)	27,208,664	33.20 GB
PubMed Central (PMC)	704,539	11.50 GB
Wikipedia	7,619,689	21.80 GB
USPTO Patent abstracts	7,996,999	7.60 GB
NCBI DailyMed	2,745	112 MB
NCBI Books	19,066	256 MB
Total	43,551,702	74.45 GB

Table 7: Statistics for BioKB's free-text document collections. This table shows the name of document collections, the number of entries in each document collection, as well as the storage size.

Structured Database	Number of Records
OMIM	23,219
T3DB	3,713
HMDB	41,513
FooDB	27,509
KEGG Reactions	9,538
KEGG Pathways	456
Gene Ontology	40,535
UniProtKB	541,561
MetaCyc	3,810
GAD	167,298
HPRD	18,863
DrugBank	6,825
Total	884,840

Table 8: Statistics for BioKB's structured database collections. This table shows the name of the database and the number of entries in each database.

Node Type	No. Nodes	No. Attributes	No. Internal Links	No. External Links
Genes/Proteins	400,303	17	372,394	350,459
Drugs	7,740	55	4,250	19,689
Drug Metabolites	1,321	27	876	1,221
Human Metabolites	41,514	64	0	1,882,510
Human Enzymes	5,688	33	0	992,105
Yeast Metabolites	2,027	54	0	22,637
Yeast Enzymes	5,158	33	0	18,328
Food Metabolites	21,239	57	0	121,210
Biological Pathways	465	15	123,656	34,007
Human Diseases	23,748	40	0	0
Toxins	4040	72	0	42,825
Toxin Targets	1,802	40	0	28,851
Biological Taxonomies	187,547	30	176,505	2,316,239
E. coli Metabolites	1594	46	0	19,667
E. coli Enzymes	6481	37	0	22,452
MeSH Terms	27,455	7	0	0
Gene Ontology Terms	41,841	8	67,478	0
Chemical Ontology Terms	4017	8	4,015	5,539
Total	783,980	643	749,174	5,877,739

Table 9: Statistics for BioKB’s knowledge graph. This table shows the name of each knowledge node, the number of node entries, the number of node attribute fields, the number of internal links (between nodes of same types), and external links (between nodes of different types).

In addition to its biomedical thesauri and document collection, BioKB also consists of an extensive biomedical knowledge graph. This knowledge graph contains more than 783,000 nodes in 18 categories with 749,000 internal links and more than 5.8 million external links. This knowledge graph is built by extracting concepts from BioKB's annotated databases or knowledgebases. More specifically, the knowledge graph's nodes are extracted from UniProt [92] (genes/proteins), DrugBank [51] (drugs and drug metabolites), HMDB [98] (human metabolites and human enzymes), YMDB [44] (yeast metabolites and enzymes), FooDB (food metabolites)[96], KEGG [47] (biological pathways), OMIM [36] (human diseases), T3DB [97] (toxins and toxin targets), Integr8 [76] (biological taxonomies), ECMDB [35] (E. coli metabolites and enzymes), MeSH [81] (MeSH terms), GeneOntology [6] (GeneOntology terms), and other in-house databases (such as ChemOnt for chemical ontology terms). Table 9 shows some of the statistics for BioKB's knowledge graph with the number of nodes and attributes as well as internal and external links regarding knowledge graph concepts.

BioKB regularly builds and updates its knowledge graph from its own large collection of high quality databases. This is done by first extracting the core concepts, synonyms, descriptions, and attributes from these databases and then identifying connecting concepts either within the same database or across different databases. BioKB downloads each source database in a flat-file text format, and then parses each database with custom parser programs capable of extracting target fields corresponding to each database-specific file format. Extracted concepts from each database are then pooled and compared to resolve internal and external links. Internal links are those explicitly referenced by the entry in the source database. For example, Citric Acid (DrugBank DB04272) is listed as interacting with aspirin, as aspirin "may increase the anticoagulant activity of citric acid". Therefore, the node on citric acid is internally linked to the node on aspirin in the knowledge graph. "Cellular tumor antigen p53" is referenced in the DrugBank [23] entry for aspirin (Acetylsalicylic acid, DB00945), therefore gene TP53 (UniProt entry P04637 P53_HUMAN) is linked to aspirin with an external link. In a scenario where a node is referencing external concepts that are not represented explicitly in the knowledge graph, we create "dummy nodes" with only concept IDs for future graph expansions. Links in the knowledge graph are directed, with the source node representing concepts described by the source database, and the target node represent concepts referenced by the source database. Each

concept on the knowledge graph is standardized to contain a list of synonyms and a description. In cases where a description or synopsis is available in the source database, BioKB extracts key sentences from the original description as a description for the concept. In cases where a description is not available or the original description is too short, BioKB generates descriptions using predefined description templates using attribute information found for the same entry. Algorithm and example templates as well as entries for generating concept descriptions are discussed in detail in Chapter 5 and Appendix A.

4.4 BioQA's Algorithms

BioQA utilizes a diverse collection of custom-developed algorithms to analyze user queries, perform concept and text snippet retrieval, transform documents and concept retrieval results and synthesize or paraphrase answers in various forms. Figure 23 shows the overall BioQA workflow, its modules/algorithms and the relationships between its various modules. When given a question in natural language text, the “Question Analysis” module analyzes the question to extract the question type, the lexical answer type, the query keywords, any association words, and contextual noun phrases. Contextual noun phrases are noun phrases that are not query keywords but can be used to enhance the search query formation. The “Query Processing” module formulates queries for both BioKB and PolySearch2. Query Processing module retrieves key concepts derived from the question using BioKB's underlying ElasticSearch [71] index. This module then generates descriptions for each available concept using the “Description Generator”, and finally extracts concept networks spanned by the concepts in the question and builds co-mentioned concept networks from the relevant documents. PolySearch2 accepts a formulated PolySearch2 query and returns a list of relevant concepts and text snippets using the PolySearch algorithm [16, 17]. Based on the query analysis, and query processing results, the “Answer Synthesis” module ranks relevant concepts, formats the concept networks, and synthesizes structured textual answers. The final textual answers are synthesized using descriptions retrieved from the “Description Generator”, as well as via summarization of relevant snippets using BioQA's greedy LSI (Latent Semantic Index) [38, 46] based summarization algorithm, or via summarization based on relationships among concepts in

the knowledge graphs. Upon user request, the “Paraphrase Module” is called to transform the initial free-text answer into a paraphrased document. Users can post their questions and view relevant concepts, knowledge graphs, and textual answers on the BioQA web interface.

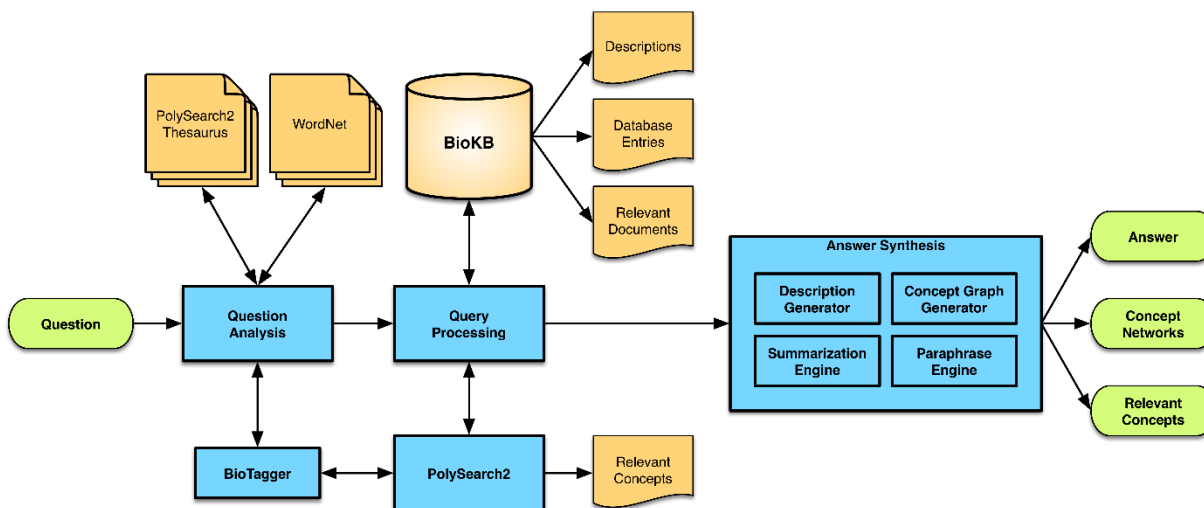


Figure 23: BioQA's knowledge and algorithmic components.

In this section, we briefly described BioQA’s algorithms for: 1) question analysis; 2) named entity recognition; 3) concept and text snippet retrieval; 4) description generation; 5) answer synthesis, and 6) automated paraphrasing. More information about BioQA’s algorithms can be found in Chapter 5 and the Appendices. Chapter 5 describes each algorithm in detail while Appendix A shows example description templates for generating descriptions from the DrugBank [51] database. Appendix B shows examples of paraphrasing rules used by the automated paraphrasing module to transform sentences into equivalent sentence forms. Finally, Appendix C describes other information extraction algorithms used by BioQA and BioKB in parsing database entries. These algorithms combine the implicit knowledge in BioKB with each user query to arrive at a final answer as seen in the BioQA web interface.

BioQA’s Question Analysis module extracts useful information from posted questions for all downstream question-answering processes. Given a question, this module: 1) identifies question types, 2) extracts Lexical Answer Type (LAT) information, 3) extracts keywords, 4)

extracts association words, and 5) extracts contextual noun phrases. This extracted information from the posted question is used to build search queries for concept and text snippet retrieval as well as for answer synthesis. BioQA supports both descriptive and associative question types. So the Question Analysis module needs to determine whether the posted question is asking for a description of certain biomedical entities (descriptive) or asking for associations between certain entities (associative). The Question Analysis module uses a rule-based system, which analyzes the question prefix to determine the type of question being asked. Another important aspect of the Question Analysis module is to extract query keywords and other elements to formulate an appropriate search query for BioKB and PolySearch2. BioQA analyzes a given question using natural language processing techniques (syntactic analysis and pattern matching) to identify query words, contextual noun phrases (non-query noun phrases), lexical answer types, and association words. The Question Analysis module tokenizes question text, performs Part-of-Speech (POS) tagging, and conducts shallow syntactic parsing to identify sentence constituents such as the subject(s), verb(s), and predicate(s). Noun phrases are also extracted from the parsed sentence using regular expression pattern matching as well as dictionary lookups using the PolySearch2 thesaurus [52], and search queries using WordNet [62]. Extracted noun phrases are assigned to the query keyword(s) or contextual noun phrases (non-query noun phrase for enhancing the specificity of the search query) based on their positions in the sentence. Lexical Answer Type (LAT) extraction is used to determine the type or format of the intended answer. For example, in a PolySearch2 query, the LAT is the type of biomedical entity we wish to find. Consider the following a few examples (with each LAT underlined): “*Which parasite causes malaria?*”, “*What diseases are associated with chemical BPA?*”. BioQA uses a rule-based method to identify the LAT in a posted question. BioQA extract noun phrases in the subject between the query prefix words and main verb of a sentence and uses predefined rules to map noun phrases to the target LAT. Verbs, adjectives, adverbs, and prepositions in the posted question are classified as association words. Using the output from Question Analysis module, The Query Processing module combines query keywords, contextual NPs, LATs, and association words to form search queries for PolySearch2 and BioKB. In particular, the Query Processing module issues formatted queries to PolySearch2 to retrieve relevant concepts with relevancy scores. This module also issues search queries to BioKB to find concept descriptions, relevant database entries, supporting documents and text snippets. Query Processing results are further

processed by the Answer Synthesis module to form natural language answers with supporting evidence. An example of a Question Analysis output for the question: “What is aspirin?” is shown in Table 10.

Input Question
What is aspirin?
Question Analysis Results
Query Keyword: Aspirin
Contextual NPs: None
Question Type: Descriptive
LAT: Aspirin

Table 10: Example Question Analysis results for the question “What is aspirin?”.

The ability for BioQA to “recognize” or “tag” biomedical entities in a given free-text question as well as entities in relevant sentence snippets is particularly important for Question Analysis, Query Processing, and Answer Synthesis. BioQA uses BioTagger, its Named Entity Recognition module, to parse noun phrases mentioning biomedical terms. BioTagger, as shown in Figure 18 and Figure 19, will take a given natural language sentence, tag biomedical entities, color code them, and hyperlink them to corresponding BioKB database entries. The BioTagger algorithm combines exact dictionary matching, shallow syntactic parsing, and N-gram language models to identify noun phrases. BioTagger first tries to match surface terms to terms in the BioKB thesauri by exact dictionary matching a concept’s synonyms against the BioKB thesauri. When no exact matching is available, BioTagger performs a combination of POS tagging, Probabilistic Context-Free Grammar (PCFG) [46] parsing, and regular expression pattern matching to extract noun phrases (NPs) that partially match terms in the BioKB thesauri. Finally, BioTagger generates frequent N-grams (for N ranging from 1 to 5) from given sentences and searches against an N-gram dataset generated using the entire MEDLINE abstract database. BioTagger prefers terms with exact dictionary matches over extracted noun phrases with partial

matching or frequent N-grams. BioTagger also prefers longer terms over shorter terms, and more frequent terms than less frequent terms. By using a number of algorithmic improvements and optimizations, BioTagger is very efficient in processing retrieved documents with a memory requirement that is linear to the size of the BioKB thesauri. Furthermore, its time efficiency is (best case) linear $O(N)$ or (worst case) $O(N^2)$ to the length of the input sentence. An example of a BioTagger output for the question: “*What is aspirin?*” is given in Table 11.

The Answer Synthesis module generates structured textual answers and augments these answers with reference citations, relevant documents and concept network diagrams using description generation, concept graph generation, automated summarization and paraphrasing. BioQA uses the Description Generator to create descriptions for concepts in a database without a description field. The Description Generator first parses a database entry for information fields according to the database’s specific schema and stores extracted fields in a lookup dictionary. It then generates descriptions by filling in the blanks using pre-defined sentence templates, thereby producing a structured description paragraph. Description templates consist of sentence templates grouped into multiple ordered sentence groups. A sentence group represents a single sentence describing one or more properties for a database entry. Each sentence group contains various hand-crafted sentence templates conveying similar information with different syntactic variations. Each sentence template contains one or more blank fields to be filled with information extracted from corresponding database entry. When all blank fields in a sentence template have the corresponding information for a database entry, these sentence templates are “triggered”. The Description Generator then randomly selects a “triggered” sentence template in the same sentence group to produce one descriptive sentence. The Description Generator module then processes each sentence group to produce the remaining descriptive sentences and join all generated sentences to produce a free-text paragraph describing the target biomedical concept. An example of a Description Generator output for the question: “*What is aspirin?*” is given in Table 12.

Input Question
What is aspirin?
BioTagger Results
<p>DB00945: Acetylsalicylic acid (DrugBank)</p> <ul style="list-style-type: none"> 130 Synonyms: Aspirin; Acetylsalicylic acid; 2-Acetoxybenzenecarboxylic acid; 2-Acetoxybenzoic acid; Azetylsalizylsäure; Acetylsalicylate; Acide acétylsalicylique; ácido acetilsalicílico; Acidum acetylsalicylicum; ASA; o-acetoxybenzoic acid; O-acetylsalicylic acid; Aspirin; o-carboxyphenyl acetate; Polopiryna; Acenterine; Adiro; Aspergum; Aspro; Bayer Aspirin; Easprin; Ecotrin; Empirin; Entrophen; Nu-seals; Rhodine; Rhonal; Solprin; Solprin acid; St. Joseph Aspirin for Adults; Tasprin; Aspirin; 2-Carboxyphenyl acetate; 8-hour Bayer; A.S.A.; A.S.A. Empirin; Acesal; Acetal; Aceticyl; Acetilsalicilico; Acetilum acidulatum; Acetisal; Acetol; Acetonyl; Acetophen; Acetosal; Acetosalic acid; Acetosalin; Acetoxybenzoic acid; Acetylin; Acetylsal; Acetylsalicylic acid; Acido O-acetil-benzoico; Acido acetilsalicilico; Acimetten; Acisal; Acylpyrin; Asagran; Asatard; Ascoden-30; Aspalon; Aspec; Aspidrops; Aspireine; Asteric; Bayer; Bayer Extra Strength Aspirin For Migraine Pain; Benaspir; Bi-prin; Bialpirina; Bialpirinia; Bufferin; Caprin; Cemirit; Claradin; Clariprin; Colfarit; Contrheuma retard; Coricidin; <p>HMDB01879: Aspirin (HMDB)</p> <ul style="list-style-type: none"> 63 Synonyms: 2-(Acetyloxy)benzoate; 2-(Acetyloxy)benzoic acid; 2-Acetoxybenzenecarboxylic acid; <p>D001241: Aspirin (MeSH)</p> <ul style="list-style-type: none"> 19 Synonyms: 2-(Acetyloxy)benzoic Acid; Acetysal; Acylpyrin; Aloxiprimum; <p>T3D2936: Aspirin (T3DB)</p> <ul style="list-style-type: none"> 26 Synonyms: 2-Acetoxybenzenecarboxylic Acid; 2-Acetoxybenzoic Acid; 2-Carboxyphenyl acetate; A.S.A.; ASA;

Table 11: Example BioTagger result for the input question “*What is aspirin?*”.

Input Question
What is aspirin?
Description Generator Result
Aspirin (USAN), also known as acetylsalicylic acid (INN, ASA), is a salicylate drug, often used as an analgesic to relieve minor aches and pains, as an antipyretic to reduce fever, and as an anti-inflammatory medication.

Table 12: Description Generator results for the question “*What is aspirin?*”

BioQA uses a Concept Graph Generator module to build a concept graph from the relevant concepts found in BioKB’s concept network as well as the co-mentioned concepts found in relevant text snippets. A concept graph is an undirected graph where vertices represent biomedical concepts and edges represent connections between concepts. Two concepts are connected by an edge if either 1) one concept references another concept in a database record, or 2) both concepts are frequently co-mentioned (above certain statistical cut-off) from the retrieved text snippets. As mentioned above, the Question Analysis module parses query keywords and noun phrases from a given question, and BioTagger maps them to a collection of corresponding biomedical concepts. To build a concept graph from concepts in BioKB’s concept network, the Concept Graph Generator module simply extracts the subgraph spanned by the collection of concepts found in the given question. In particular, the extracted subgraph includes: nodes representing the query term, biomedical terms representing the contextual noun phrases, their immediate neighbors and all connecting edges. To build a concept graph from co-mentioned concepts in retrieved text snippets, the Concept Graph Generator module scans each relevant sentence and extracts pairs of biomedical concepts found in a same sentence. It also keeps a numerical count of the frequency of each concept pair in a dictionary data structure. The dictionary represents an implicit background concept network, where each pair of co-mentioned concepts is connected by an edge, with the strength of an edge being proportional to the frequency of it being co-mentioned in a sentence. Next, the module trims this implicit background concept network by scoring and ranking each co-mentioned concept pair using Z-

distribution statistics. Edges with a Z-score lower than 0 (therefore the observed association is likely due to chance) are removed from the implicit background network. Finally, the Concept Graph Generator module extracts subgraphs from the trimmed background network by including only those nodes and edges connecting to concepts found in the given question. Both concept networks can be visualized in the BioQA user interface as shown in Figure 21 and Figure 22. An example of a Concept Graph Generator output for the question: “What is aspirin?” is given in Table 13.

Source ID	Source Node Type	Source Node Name	Target Node ID	Target Node Type	Target Node Name
DB00945	Drug	Acetylsalicylic acid	C081124	Compound	acetyl chloride
DB00945	Drug	Acetylsalicylic acid	D014481	MeSH	United States
DB00945	Drug	Acetylsalicylic acid	DB00316	Drug	APAP
C081124	Compound	acetyl chloride	DB00936	Drug	Salicyclic acid
D014481	MeSH	United States	D014486	MeSH	United States Food and Drug Administration
D014481	MeSH	United States	D047828	MeSH	World War I
D014481	MeSH	United States	DB00497	Drug	Oxycodone
DB00316	Drug	APAP	HMDB01859	Human Metabolite	4'-Hydroxyacetanilide
DB00316	Drug	APAP	C526278	Compound	acetaminophen, codeine drug combination
DB00316	Drug	APAP	C019552	Compound	Saridon

Table 13: Example of a Concept Graph Generator output on the input question “What is aspirin?”. A subset of 10 edges in the concept graph (51 nodes, 44 edges) are shown in this table. This table shows the concept ID, node type, and node name for source and target nodes for selected edges.

Input Question
What is aspirin?
Summarization Engine Output
<p>Aspirin (USAN), also known as acetylsalicylic acid (INN, ASA), is a salicylate drug, often used as an analgesic to relieve minor aches and pains, as an antipyretic to reduce fever, and as an anti-inflammatory medication. Although aspirin 's use as an antipyretic in adults is well-established, many medical societies and regulatory agencies (including the American Academy of Family Physicians, the American Academy of Pediatrics, and the U.S. Food and Drug Administration (FDA)) strongly advise against using aspirin for treatment of fever in children because of the risk of Reye 's syndrome, a rare but often fatal illness associated with the use of aspirin or other salicylates in children during episodes of viral or bacterial infection. After the association between Reye's syndrome and aspirin was reported, and safety measures to prevent it (including a Surgeon General 's warning, and changes to the labeling of aspirin-containing drugs) were implemented, aspirin taken by children declined considerably in the United States, as did the number of reported cases of Reye's syndrome; a similar decline was found in the United Kingdom after warnings against pediatric aspirin use were issued. The company's attempts to hold onto its Aspirin sales incited criticism from muckraking journalists and the American Medical Association, especially after the 1906 Pure Food and Drug Act that prevented trademarked drugs from being listed in the United States Pharmacopeia; Bayer listed ASA with an intentionally convoluted generic name (monoacetic acid ester of salicylic acid) to discourage doctors referring to anything but Aspirin. Surgeon General, the Food and Drug Administration, the Centers for Disease Control and Prevention, and the American Academy of Pediatrics recommend that aspirin and combination products containing aspirin not be given to children under 19 years of age during episodes of fever-causing illnesses, because of a concern about Reye's Syndrome.</p>

Table 14: Summarization engine output for the question “What is aspirin?”.

Input Question
What is aspirin?
Paraphrasing Engine Output
<p><u>Acetysal</u> (USAN), also known as <u>acetylsalicylic acid</u> (INN, <u>argininosuccinic acid</u>), is a <u>salicylate</u> drug, often used as an analgesic to relieve nonaged aches and pains, as an <u>antipyretic</u> to reduce <u>hyperthermias</u>, and as an <u>anti-inflammatory</u> medication. Although <u>acetysal</u>'s use as an <u>antipyretic</u> in adults is well-established, many <u>medical societies</u> and regulatory agencies (including the American Academy of <u>extended Family</u> Physicians, the American Academy of Pediatrics, and the U.S. <u>Food and Drug Administration</u> (FDA)) strongly advise against using <u>acetysal</u> for therapeutics of fevers in children because of the comparative Risks of Reye's <u>symptom Cluster</u>, a rare but often fatal unwellness associated with the use of <u>acetysal</u> or other <u>salicylates</u> in children during episodes of viral or <u>bacterial Infections</u>. After the associations between Reye's <u>symptom Cluster</u> and <u>acetysal</u> was reported, and refuge measures to prevent it (including a Surgeon General's warning, and changes to the labeling of acetysal-containing drugs) were implemented, <u>acetysal</u> taken by children declined well in the <u>United States</u>, as did the figure of reported cases of Reye's <u>symptom Cluster</u>; a similar diminution was found in the United Kingdom after warnings against paediatric <u>acetysal</u> use were issued. The company's attempts to hold onto its <u>acylpyrin</u> sales incited unfavorable judgment from muckraking journalists and the association, American Medical, especially after the 1906 Pure Food and Drug Act that prevented trademarked drugs from being listed in the <u>United States</u> Pharmacopeia; acetyl2-Hydroxybenzoic Acid listed <u>argininosuccinic acid</u> with a deliberately convoluted generic name (monoacetic acid ester of <u>2-Hydroxybenzoic Acid</u>) to discourage doctors referring to anything but <u>acylpyrin</u>. Surgeon General, the foods and Drug Administration, the <u>Centers for Disease Control</u> and Prevention, and the American Academy of Pediatrics recommend that <u>acetysal</u> and combining products containing <u>acetysal</u> not be given to children under 19 years of historic period during episodes of fever-causing illnesses, because of a care about Reye's <u>symptom Cluster</u>.</p>

Table 15: Example Paraphrasing Engine output for synthesized answers with input question “What is aspirin?”.

BioQA synthesizes its answers in natural language text using the Summarization Engine module. The Summarization Engine combines concept descriptions retrieved from BioKB or generated using the Description Generator, with answer paragraphs describing the association between relevant concepts. The Summarization Engine “composes” its answers using two different algorithms: a) Summarization by Co-mentioned Concept Graph, that is generating a summary paragraph from co-mentioned concept information, and 2) Summarization by Greedy LSI, that is generating a summary paragraph from relevant text snippets using a greedy algorithm on a Latent Semantic Index data structure for relevant documents. The first algorithm (Summarization by Co-mentioned Concept Graph) takes advantage of the co-mentioned concept graph (built by the Concept Graph Generator module) and the interconnectivity of relevant concepts. Synthesizing answers using a co-mentioned concept graph involves a form of implicit reasoning, where the algorithm joins sentences describing entity connections across multiple linked concepts in the natural order found in the relevant text snippets. In other words, to generate a paragraph describing the association between two concepts X and Y, this algorithm first finds a shortest path (if any) in the co-mentioned concept graph using the Single Source Shortest Path algorithm (e.g. a shortest path connecting concept X and Y could be X-Z-Y, where concept Z connects both X and Y) [20]. Then the algorithm traverses each pair of concepts in the shortest path (e.g. X-Z, and Z-Y) from the source concept X to the target concept Y, and selects the highest ranked sentence (based on PolySearch2’s relevancy score) from all sentences containing both concept X and concept Y. The strength of this algorithm is that only sentences containing the strongest evidence for the association between the two concepts are included in the final summary. The weakness of this algorithm is that a path between two concepts may not exist in co-mentioned concept graph. In this case, a default summary is generated using the second algorithm (Summarization by Greedy LSI), where the given question “grows” to a summary paragraph with the help of a latent semantic index from the relevant documents. Synthesizing answers using a document matrix with a latent semantic index involves information filtering to identify key terms and key sentences among all relevant text snippets. The second algorithm (Summarization by Greedy LSI) first builds a Latent Semantic Index (LSI) from the retrieved relevant documents. Then starting from the given question sentence it greedily includes the next most similar sentence to the current summary. Given a question and a collection of relevant documents (or text snippets), this algorithm converts each relevant snippet

to a document vector representation and then forms a document matrix via a vector space model [46, 56]. It then calculates eigenvectors and eigenvalues of the document matrix using Singular Value Decomposition (SVD) and reduces the dimension of the document matrix by projecting document vectors onto a lower dimension space spanned by the eigenvectors. The eigenvectors of the document matrix represent key topics (biomedical terms) found among relevant documents. Therefore, this dimensional reduction step effectively filters key topics among the collection of relevant snippets and indexes each text snippet with key terms. Finally, the algorithm greedily generates a summary paragraph using the initial question document vector and the LSI document index in subsequent iterations. That is, given an initial question document vector, the algorithm retrieves text snippets corresponding to the most similar document vector in the document index by Cosine Similarity measure. The algorithm adds the retrieved snippets to the summary paragraph, removes snippets similar to the current snippet above an empirical threshold, and recalculates the document index, now containing fewer documents. This process is repeated until the summary paragraph grows to a certain length, or the document matrix contains too few relevant snippets to continue the indexing process. Finally, the Summarization Engine performs post-processing on the generated summary paragraph to enhance readability and fixes grammatical artifacts (introduced during summarization) to produce the final summary paragraphs. An example of a Summarization Engine output generated using the two different algorithms in this Engine for the question: “*What is aspirin?*” is given in Table 14.

The answers that BioQA generates are almost always composed of previously existing text that may or may not be copyrighted. Therefore, BioQA also supports automated paraphrasing of natural language answers for those users who wish to include all or part of BioQA’s answer in a document without the need to manually paraphrase the answer. The Paraphrasing Engine module takes an initial BioQA textual answer, and paraphrases it, sentence by sentence according a set of pre-defined rules. Paraphrasing rules falls into substitution, enumeration, rearrangement, and transformation categories. Please refer to Appendix C for more details and examples regarding these paraphrasing rules. The Paraphrasing Engine module applies phrase substitution, word-sense substitution, and synonym substitution to an input sentence. In particular, this module applies 2000+ phrase or word substitution rules (see Appendix B) to an input sentence to replace a phrase with its semantic equivalent. These

substitution rules can be simple or word-sense dependent (substitution rules depends on the Part-of-Speech tags for the original words). Simple substitution replaces a phrase with an equivalent phrase. For example, substituting “also known as” with “also referred to as”. Word-sense substitution switches a word based on its Part-of-Speech tag. For example, the word “witness” can be substituted with “observe” when “witness” is used as a verb, but with “observer” when “witness” is used as a noun. The paraphrase engine then substitutes a word with a valid synonym by searching WordNet [62] (English dictionary words) and the BioKB thesaurus (biomedical terms). The module recognizes phrases or common expressions such that synonyms substitution does not replace a part of a phrase or common expression by mistake. Besides word substitutions, this module also performs transformations, enumerations, and rearrangements to paraphrase an input sentence. Transformation rules changes a numerical measure to an equivalent with different units. Rearrangement rules rearrange words in an expression. In paraphrasing, the module also obeys other rules that don’t easily fit into the previous categories. For example, it should never change anything in quotes, and never change proper nouns, acronyms (“BPA”) or entity names (“Bisphenol A”). When multiple rules are applicable to an input sentence, there could be a potential conflict between rules, as more than one rule could be substituting the same part of the sentence yielding different results. In this case, only one rule is selected among the conflicting rules (according to predefined rule precedence or at random) to paraphrase a sentence. Besides handling conflicting rules, The Paraphrasing Engine also randomizes paraphrasing results to a certain degree to provide a higher degree of syntactic variance. Running the paraphrasing function over and over again should yield a slightly different paragraph (but with the same meaning) each time it is run. The output from the Paraphrasing Engine is a paraphrased version of the original answer/paragraph with original references. An example of a Paraphrasing Engine output (before and after running the Engine) for the question: “*What is aspirin?*” is given in Table 15.

In this section, we briefly discussed the various algorithms that BioQA uses for performing question analysis, query processing, and answer synthesis. Answer synthesis utilizes an algorithm to generate concept descriptions from database entries using predefined templates. It also uses an algorithm to build relevant concept graphs from BioKB’s concept network and co-mentioned concept graph. It then uses an algorithm to automatically summarize concept

associations using the concept graph or document matrix, and an algorithm to automatically paraphrase the answer. Working as a whole, these algorithms and modules transform an input question into answers in a variety of different forms, including natural language answers with reference citations, a ranked list of relevant concepts, and an image of the relevant concept graphs.

4.5 Performance Evaluations

In Chapter 3 we previously evaluated the performance of PolySearch2 [8] with regard to its information retrieval capacity and sensitivity. In this section we present two different, independent evaluations on BioQA with regard to its Query Processing and Answer Synthesis components. We first evaluate BioQA's question analysis module by evaluating its question type identification capability. Next, we evaluate BioQA's answer synthesis modules using the BioASQ challenge dataset [82, 88, 89].

4.5.1 Question Analysis Evaluation

A key aspect for any free-text question-answering system is its ability to accurately identify the type of input question being asked. In most cases it is a matter of distinguishing if a question is descriptive or associative. We evaluated BioQA's question type identification algorithm using the BioASQ's training dataset [82, 88], which consisted of 600 questions with specific question types: 1) yes/no, 2) descriptive, 3) associative, and 4) summary. In this evaluation we classify summary questions as associative question types. We compare BioQA's prefix rule-based algorithm with three other commonly used algorithms: 1) a K-nearest neighbor algorithm 2) a Support Vector Machine classifier, and 3) a Random Forest classifier. The K-nearest neighbor algorithm is an instance-based learning algorithm, and is often used to evaluate classification problems as a baseline system due to its simplicity. Our K-nearest neighbor algorithm takes the majority of the top 3 questions that are most similar to the given question in the BioASQ training dataset. If the top 3 questions are of different types, we take the question type belonging to the most similar question. Support Vector Machine and Random Forest

classifiers are both popular choices for classification tasks in text mining due to their excellent capacity to handle high dimensional features. In all three classification algorithms, we used stemmed bag-of-words features vectors, and Term Frequency / Inversed Document Frequency (TF-IDF) term weighting schema [46, 56]. We use cosine similarity as similarity measure between feature vectors. Note that in this evaluation, stop-words are not removed as stop words can be valuable features for question type prediction. We evaluated all three supervised classification algorithms using 5-fold cross validation, and compared their performance with the prefix rule-based algorithm used in BioQA. As seen in Table 16, BioQA's prefix rule algorithm outperforms all three supervised classification algorithms. These data illustrate that a supervised classification algorithm may not be a better option in question type analysis than a system based on hand-crafted empirical rules for this kind of task. We speculate that for supervised classification algorithms to achieve the desired accuracy, we would need a much larger training dataset. This is likely due to the fact that questions of same type may not share enough words or features to sufficiently characterize a specific type of question. Rule-based systems that uses hand-craft empirical rules, on the other hand, examine question prefixes based on empirical rules, and so they are less prone to the size of training dataset.

To test this hypothesis, we conducted the following experiment: we converted each question in the BioASQ training dataset into a feature vector with stemmed bag-of-words features. Similar to our evaluation, stop words were preserved in the feature vectors. We weighted each feature using TF-IDF, and measured the cosine similarity between each question and the most similar question (excluding itself) among all three question types (yes/no, associative, descriptive). We visualized the similarity scores using a series of scatter plots (Figure 24, Figure 25, and Figure 26). Each question can be visualized as a data point on a scatter plot. The location of each data point corresponds to its maximum similarity score to a question type along an axis. If instances of each question type are well separable based on their feature vectors, we should see questions of same type cluster into relatively distinct clusters. Based on the data in Figure 24, Figure 25, and Figure 26, we see that this is not the case. In particular, there is little clustering, indicating question types may not be easily predicted using linear machine learned classifiers. Based on this observation, and the overall superior performance, BioQA uses an empirical rule-based question type identification algorithm.

Question Type	Performance Measure	Prefix Rule	KNN	SVM	Random Forest
Yes/No	Precision	0.9241	0.4233	0.5541	0.5468
	Recall	1.00	0.4533	0.2482	0.2165
	Accuracy	0.9769	0.6733	0.7174	0.7289
	F-measure	0.9605	0.4378	0.3428	0.3102
Associative	Precision	0.9438	0.6678	0.6891	0.7558
	Recall	0.6981	0.6294	0.5344	0.5138
	Accuracy	0.8291	0.6598	0.6273	0.6748
	F-measure	0.8026	0.6483	0.6028	0.6117
Descriptive	Precision	0.5811	0.4655	0.6394	0.6809
	Recall	0.8602	0.4839	0.2142	0.2712
	Accuracy	0.8315	0.7623	0.7832	0.8081
	F-measure	0.6936	0.4745	0.3209	0.3879

Table 16: Performance statistics of BioQA’s question type prediction algorithm (prefix rule) in comparison with K-nearest neighbor (KNN), Support Vector Machine (SVM), and Random Forest classifiers on the BioASQ training dataset with 600 questions with question type labels.

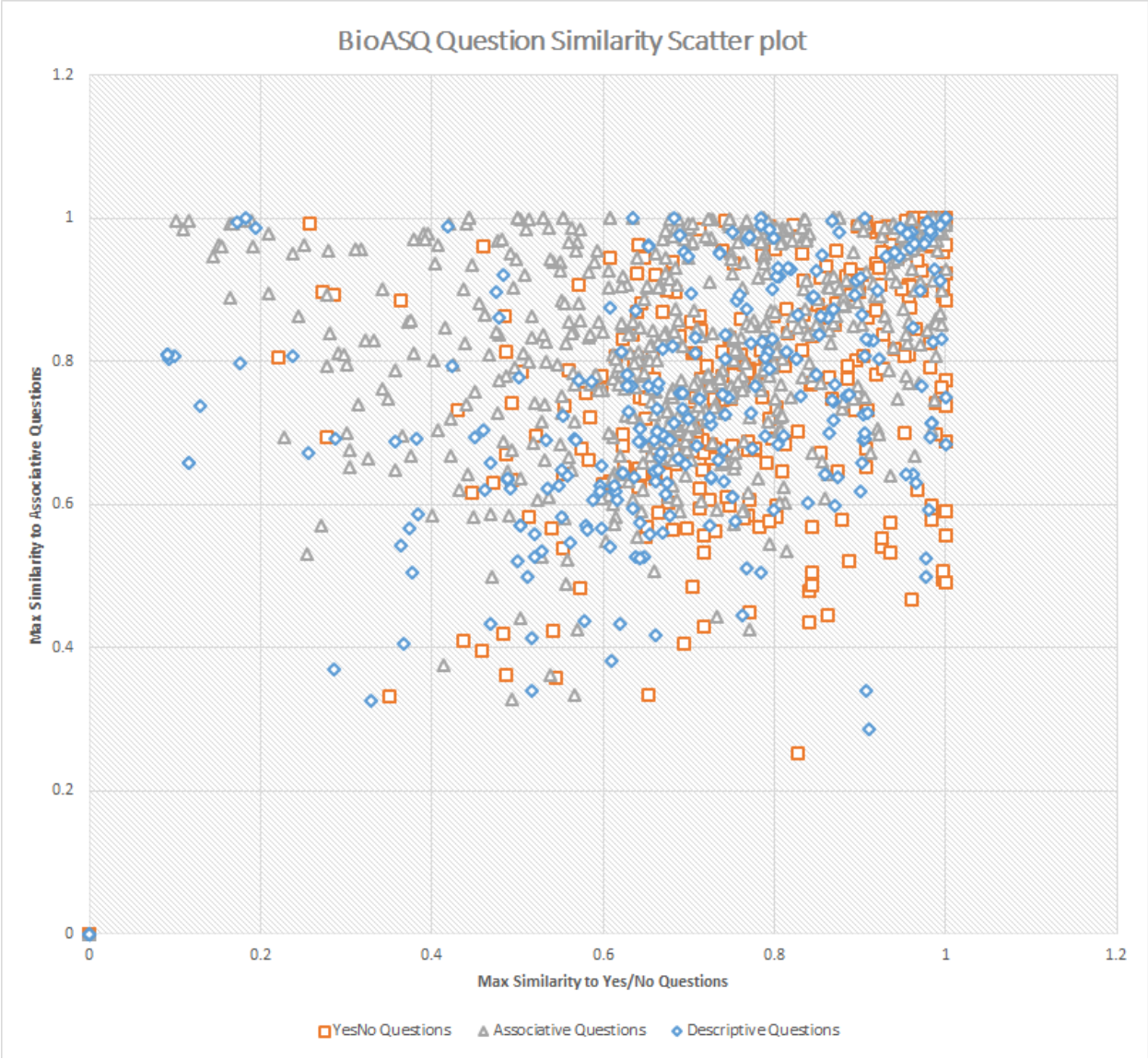


Figure 24: BioASQ Question Similarity Scatter plots: Yes/No questions versus Associative questions.

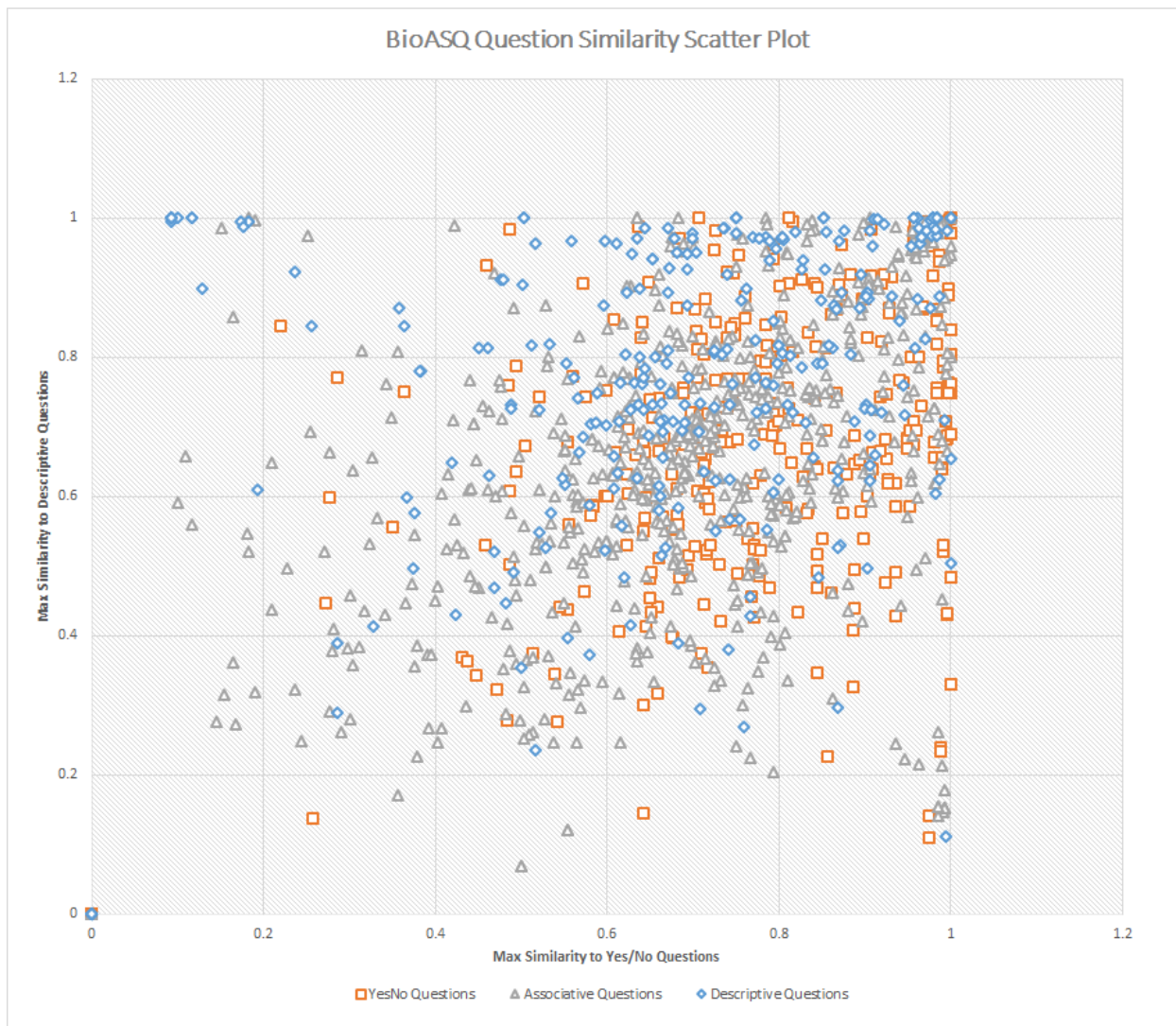


Figure 25: BioASQ Question Similarity Scatter plots: Yes/No questions versus Descriptive questions.

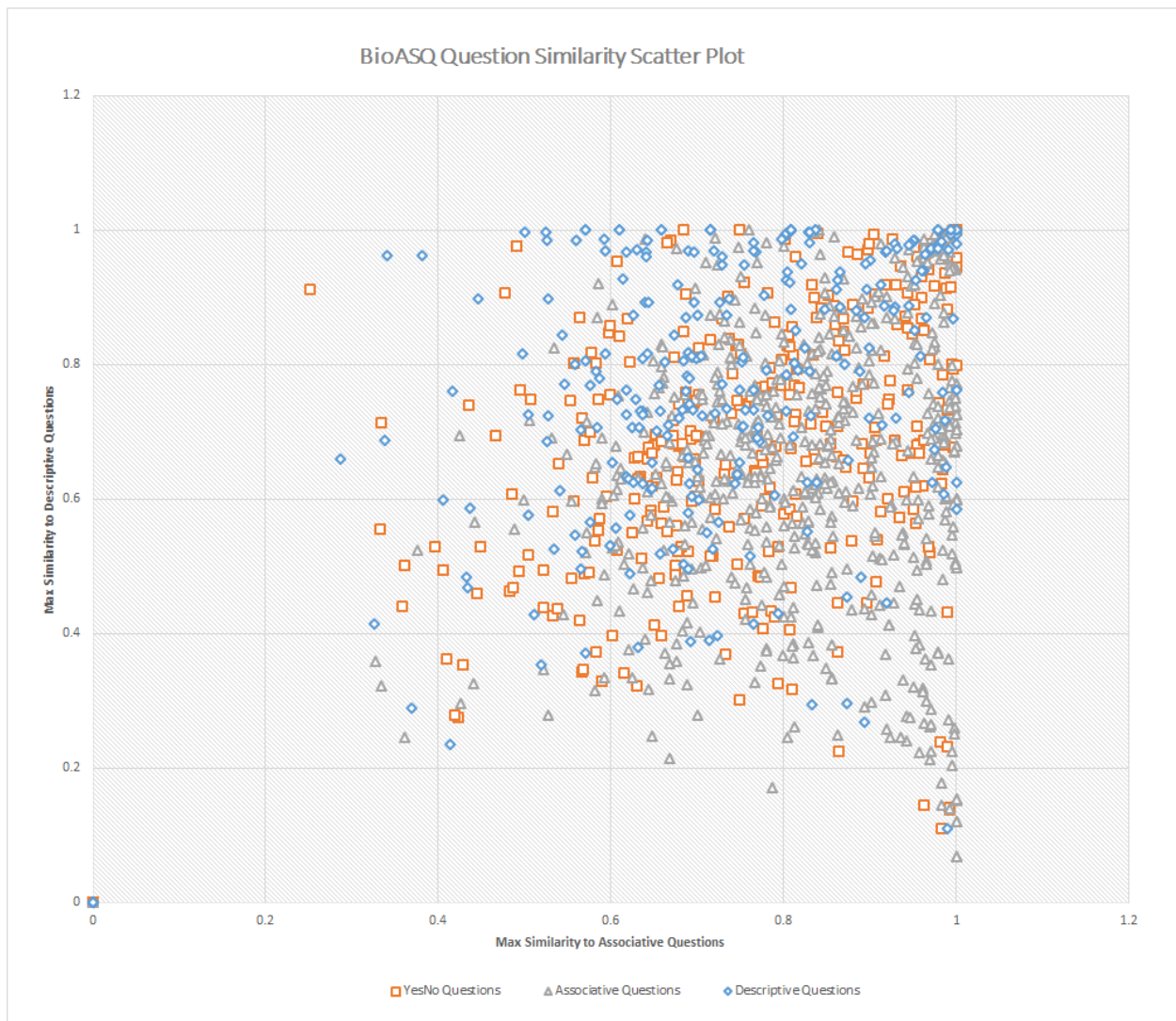


Figure 26: BioASQ Question Similarity Scatter plots: Associative question versus Descriptive questions.

4.5.2 Answer Synthesis Evaluation

To assess the overall performance of BioQA in terms of question answering, we evaluated it by participating in the first and second BioASQ challenges. The first BioASQ challenge took place in 2013, and the second BioASQ challenge took place in 2014. A total of 11 participants were involved in BioASQ-1 (2013) and 15 participants in BioASQ-2 (2014) for Task B. BioASQ (<http://bioasq.org>) [82, 88] is a semantic question answering challenge with two distinct tasks. Task A challenges participants to automatically index novel MEDLINE abstracts with MeSH tags; Task B challenges participants to annotate given natural language questions with relevant articles, text snippets, and RDF triples from designated document and concept repositories (Phase A), and eventually return an “exact” and an “ideal” answer in natural language (Phase B). Participants are allowed to process a challenge question set and submit answers within 24 hours. Submission results are evaluated automatically and manually by a panel of biomedical experts. Please refer to Satsaronis et al. [82, 88] and Malakasiotis et al. [55] for details on the BioASQ challenge and evaluation measures. Specifically, we evaluated BioQA’s modules in both Task A and Task B with modifications in order to comply with challenge guidelines. The information retrieval module in BioQA was temporarily customized to retrieve information from the BioASQ article and concept repository instead of from BioQA’s local document and concept repository (BioKB). The Question Processing module was customized to accept natural language questions with given question types. The Answer Synthesis module was customized to process relevant articles, snippets, and concepts (provided by BioASQ) plus information retrieved locally from BioKB and PolySearch2 (e.g. concept descriptions and associations), to synthesize the final “exact” and “ideal” answers. Note that the version of BioQA used in the first BioASQ challenge was Version v1.1 while the version of BioQA used in the second BioASQ challenge was Version v1.2. Version v1.1 is equipped with core algorithms for question analysis and answer synthesis. Version v1.2 exploits an algorithm for greedily removing sentences with redundant information and “smoothing” sentence transition within an answer paragraph by rearranging sentences based on their information connection. The current version of BioQA is Version v1.3. Version v1.3 is enhanced with public web interface, concept graph visualization, and automated paraphrasing. Performance evaluation presented in this section is based on the participation of BioQA version v1.1 in the first BioASQ challenge,

and version v1.2 in the second BioASQ challenge. Challenge results are publicly available on the BioASQ challenge website (<http://bioasq.org>), and are discussed in Partalas et al. [72] and Balikas et al. [12] with references to BioQA as the “Wishart” systems. Performance statistics including precision, recall, accuracy, mean reciprocal rank (MRR), and F-measure and accuracy are presented in Table 17 for exact answer formation in six evaluation runs. Automatic and manual evaluation scores are presented in Table 18 for ideal answer formation in same six runs. Performance evaluations are also available on the evaluation page of the BioQA web server. BioQA’s performance in concept retrieval were evaluated with PolySearch2 (<http://polysearch2.ca>) [52]. In this section, we focus on discussing BioQA’s ability in formulating exact and ideal answers. We used a two-sample t-test to compare key performance statistics of BioQA against the performance statistics of the best system among other BioASQ participants.

		Yes/No	Factoid			List		
Exact answer		Acc.*	Strict Acc.	Lenient Acc.	MRR*	Mean Prec.	Recall	F-measure*
Task 1b Phase B Batch 1	BioQA	0.9200	0.2222	0.3333	0.3056	0.3186	0.2147	0.2290
	Other Best	0.4800	0.0000	0.2222	0.1056	0.0153	0.0402	0.0204
Task1b Phase B Batch 2	BioQA	0.9615	0.2500	0.3000	0.3000	0.4060	0.3127	0.3336
	Other Best	0.5000	0.0000	0.2500	0.0725	0.0612	0.2062	0.0789
Task2b Phase B Batch1	BioQA	0.8438	0.4400	0.4800	0.4600	0.4478	0.3335	0.3456
	Other Best	0.9375	0.1600	0.1600	0.1600	0.0572	0.0702	0.0614
Task2b PhaseB Batch2	BioQA	0.9286	0.1304	0.1304	0.1304	0.5120	0.4399	0.4261
	Other Best	0.8214	0.0435	0.1739	0.0942	0.1596	0.2057	0.1618
Task2b PhaseB Batch3	BioQA	0.8889	0.0417	0.0833	0.0556	0.4584	0.3763	0.3909
	Other Best	0.8333	0.0417	0.1250	0.0833	0.1195	0.1780	0.1373
Task2b PhaseB Batch4	BioQA	0.9375	0.2500	0.2813	0.2813	0.2659	0.4029	0.2963
	Other Best	0.8750	0.0625	0.1875	0.1120	0.1233	0.1365	0.1062
Overall Average	BioQA	0.9133 ±0.0415	0.2224 ±0.1342	0.2681 ±0.1438	0.2555 ±0.1432	0.4015 ±0.0926	0.3467 ±0.0793	0.3369 ±0.0696
	Other Best	0.7412 ±0.1989	0.0513 ±0.0589	0.1864 ±0.0446	0.1046 ±0.0307	0.0894 ±0.0535	0.1395 ±0.0707	0.0943 ±0.0516

Table 17: BioASQ Challenge Task B Exact answer formation. This table shows the performance statistics for BioQA v1.1 in Task1b, and BioQA v1.2 in Task 2b. Stric Acc. and Lenient Acc. stands for Strict Accuracy, and Lenient Accuracy respectively. MRR stands for mean reciprocal rank. Official ranking measures for each answer category are marked with asterisks. Those measures for which BioQA’s overall performance was significantly better than the best among other participants are shown in **bold**.

Ideal answer		Automatic Scores		Manual Scores			
		Rouge-2	Rouge-SU4	Readability	Recall	Precision	Repetition
Task1b PhaseB Batch1	BioQA	0.2059	0.2202	3.97	3.71	3.83	4.27
	Other Best	0.2266	0.2636	2.55	3.15	2.54	3.21
Task1b PhaseB Batch 1	BioQA	0.2106	0.2387	4.14	4.14	4.17	4.48
	Other Best	0.2204	0.2659	2.92	3.87	3.08	3.50
Task2b PhaseB Batch1	BioQA	0.4802	0.4814	-	-	-	-
	Other Best	0.4971	0.4971	-	-	-	-
Task2b PhaseB Batch2	BioQA	0.3914	0.4089	-	-	-	-
	Other Best	0.3352	0.3493	-	-	-	-
Task2b PhaseB Batch3	BioQA	0.4331	0.4427	-	-	-	-
	Other Best	0.4282	0.4386	-	-	-	-
Task2b PhaseB Batch4	BioQA	0.4072	0.4295	-	-	-	-
	Other Best	0.3273	0.3677	-	-	-	-
Overall Average (Standard Deviation)	BioQA	0.3547 ± 0.1174	0.3702 ±0.3952	4.055 ±0.1202	3.93 ±0.30	4.00 ±0.24	4.38 ±0.15
	Other Best	0.3391 ± 0.1094	0.3637 ±0.0930	2.7400 ±0.2600	3.51 ±0.51	2.81 ±0.38	3.36 ±0.21

Table 18: BioASQ Challenge Task B Ideal answer formation. This table shows performance statistics for BioQA v1.1 in Task1b, and BioQA v1.2 in Task 2b. Manual scores for Task 2 were not available. Those measures for which BioQA’s overall performance scores were significantly better than the best among other participants are shown in **bold**.

In the task of formulating exact answers, BioQA responded to each question by providing “yes” or “no” answers to Yes/No questions, a list of at most five concepts to factoid questions, and a list of at most 100 concepts to list questions. Yes/No questions were evaluated with an accuracy score, while factoid questions were evaluated by Mean Reciprocal Rank (MRR), which rewards responses containing golden answers (provided by experts) higher in the returned list of factoids. List questions were evaluated using standard precision, recall, and F-measure scores averaged over all submitted responses. As seen in Table 17, BioQA achieved significantly higher accuracy [0.8438 – 0.9615] in Yes/No questions ($t = 2.076$, $df = 5$, $p = 0.046$), significantly higher mean reciprocal ranks [0.0556 – 0.4600] in Factoid questions ($t = 2.5229$, $df = 5$, $p = 0.0265$), and significantly higher F-measure [0.2290 – 0.4261] ($t = 6.8542$, $df = 5$, $p = 0.0001$) in list questions than the best system among other participants. This result shows that BioQA is quite effective in formulating exact answers. BioQA’s performance in formulating exact answers can be attributed to the performance of the named entity recognition and concept ranking algorithm in PolySearch2 [52].

In the task of formulating ideal answers (Table 18), BioQA responded to each question by synthesizing a natural language text answer with at most 200 words. Submitted responses were evaluated using both automatic scores (Rouge-2 and Rouge-SU4) and manual scores (readability, recall, precision, and repetition). The automatic scoring schemes (Rouge-2 and Rouge-SU4) measure overlap ratios between the submitted summary and a set of “gold standard” summaries curated by biomedical experts using skip bigrams (Rouge-2) and skip unigrams (Rouge-SU4). Manual scores evaluate the readability, recall (concepts in the gold standard also occurs in the submitted answer), precision (concepts in the submitted answer also occur in the gold standard), and repetition (lack of repeating the same concepts in the submitted answer). Manual score ranges from 1 (worst) to 5 (best) and are assigned manually by biomedical experts. Readability scores assess how readable a summary is in terms of its content, grammar and style. Precision scores and Recall scores are not traditional precision and recall measures, as they assess, using a score from 1 to 5, how much information is shared between the submitted summary and the set of reference summaries, in comparison with the submitted summary (precision score), or the set of reference summaries (recall score). Finally, the Repetition score assesses the submitted summary for lack of repetition of the same concepts or text snippets. A

higher repetition score indicates the submitted summary contains less repeating information, and therefore is a better answer. In this evaluation, BioQA achieves moderate performance in automatic scores (Rouge-2 = [0.2059 – 0.4802], Rouge-SU4 = [0.2202 – 0.4814]) in comparison with other participating systems. None of BioQA’s automatic scores proved to be significantly higher than the best systems among other BioASQ participants. However, when evaluated by biomedical experts with manual scores ranging from one (worst) to five (best), BioQA achieved a significantly higher Readability score [3.97 - 4.14] ($t = 6.4835$, $df = 2$, $p = 0.0487$), Recall score [3.83 – 4.17] ($t = 3.7297$, $df = 2$, $p = 0.0325$), and Repetition score [4.27 – 4.48] ($t = 5.6975$, $df = 2$, $p = 0.0147$) than the best system among other participants. BioQA achieved a moderately higher (not statistically significant) Precision score [3.71 – 4.14] in comparison with other systems. Comparing two versions of BioQA, v1.2 achieves higher automatic scores (ROUGE-2 and ROUGE-SU4) thanks to the few enhancements implemented in the answer synthesis module, which reduces redundant information in the final answer, therefore leaving space to more relevant information, and leading to higher ROUGE-2 and ROUGE-SU4 scores.

BioQA’s relatively high readability score can be attributed to BioQA’s Summarization Module with “information smoothing” enhancements. That is, to ensure a smoother transition between sentences, BioQA selects a subsequent sentence in the summary based on currently selected sentences, avoids sentences starting with anaphor (pronouns referring to information in previous sentence), and actively rearranges selected sentences to achieve a better transition between sentences. BioQA’s high recall and precision scores can be attributed to the Summarization Module’s strict sentence selection techniques. The high repetition score (lack of repeated information) can be attributed to BioQA’s Summarization Module which was optimized to reduce repeating information. To avoid including repeating information in its final summary, BioQA only selects a single key sentence from a set of sentences describing an association from a graph-based summarization, and it actively trims sentences containing similar information in its sentence matrix-based summarization. Furthermore, avoiding repeated information appears to improve the recall score, as the submitted summaries are limited to a maximum of 200 words. Therefore, including less repetitive or redundant information means BioQA can deliver a more comprehensive summary thereby increasing its chances of overlapping with the “gold standard” reference answers. The above result shows that BioQA’s summaries are relatively easy to read,

contain more information (as measured against gold-standard answers), and contain less redundant information than any other biomedical question-answering system. BioQA also achieves a well-balanced performance that takes into account both accuracy and readability when synthesizing its answer summaries. However, BioQA is not perfect. All four manual scores for its answer quality are still around 4.0, which suggests there are still room for improvement before BioQA can achieve a satisfying performance (overall scores of 5 across all four manual scores) to rival human experts.

4.6 Limitations and Future Plans

No question answering system is perfect and BioQA certainly still has plenty of room for improvement. One major limitation is that BioQA is not yet capable of adapting to a specific annotation needs. Search engines like Google and Bing monitor user search activity through search-log mining and web click frequencies. These can be used to create better rankings, and provide more personalized searches by considering a user's previous search history. Currently, BioQA is a state-less machine, meaning that it treats each search query as a brand new query and does not make reference to previous searches. However, a natural use-case scenario for BioQA could be that user progressively asks more specific questions through a sequence of related searches. Another scenario is that different users may have different needs and some users may favor precision over recall while others refer the opposite. We could enhance BioQA to be an "adaptive QA" system that constantly improves its answers based on previous query submissions (from the same user or during the same search session). In order to adapt to an individual, BioQA could be modified to automatically build a custom collection of search keywords from a user's previous searches, and use this keyword list to adjust ranking scores for retrieving concepts, text snippets, and answer synthesis. By providing a way for users to rank returned answers it may also be possible to help BioQA better adapt to a user's specific needs. By keeping track of a series of questions asked by the same user and taking user feedback into consideration for subsequent searches, BioQA could progressively improve upon itself and adapt to individual user needs. Over the long term, through the use of the web's adaptive monitoring tools, BioQA

could evolve to be an adaptive and conversational QA engine that delivers answers to users through a sequence of human-machine dialogues.

Another limitation to the current version of BioQA is that it is unable to perform logical inference. BioQA addresses the information needs of users by automatically parsing user questions, searching for relevant information, and synthesizing textual answers with references. The current BioQA framework is solely based on information retrieval, text snippet extraction and statistical summarization. It lacks the capacity to perform logical or semantic reasoning. This limitation is partly due to a lack of available logical knowledge representations in the biomedical domain. Modern QA systems are moving towards reasoning and semantic processing to enhance their question answering capabilities and user experience. For example, IBM Watson supports a certain degree of semantic reasoning through the use of semantic “frames” that encapsulate semantic relations. For example, Watson [28] is able to answer question about capital city in a country without the need to perform an extensive text search and summarization. Knowledge Engines such as Wolfram Alpha [99] support certain logical reasoning operations like solving simple mathematical equations. In order for BioQA to support reasoning, it needs to convert user questions to more than just search queries, but also to logical representation to validate against a collection of logical entailments representing existing biomedical knowledge. Knowledge resource equivalents to FrameNet [11] are still scarce in the biomedical domain. BioQA’s biomedical concept network is a first step towards building a biomedical “FrameNet” that captures explicit relations between biomedical entities. BioQA can also take advantage of domain knowledge available within a smaller subfield. For example, within certain subfields, highly structured or curated information exists. For example, KEGG captures knowledge on biochemical pathways and reactions between chemicals and enzymes. Therefore, KEGG can be used to make certain inferences on biochemical pathways. In this regard it may be possible to build and attach specific “inference engines” with subfield-specific knowledge to BioQA, thereby enhancing its question answering capability with basic inference capabilities.

4.7 Conclusion

In this chapter, we introduced BioQA, a novel, high-performance biomedical question-answering system. BioQA contributes to the field of biomedical question-answering by introducing an end-to-end QA framework focusing on answering common biomedical questions. The task of question answering presents two unique challenges: knowledge representation and knowledge transformation. The former deals with knowledge acquisition and representation. BioQA solves this component by making use of its unique and extensive knowledgebase – BioKB. BioKB encapsulates large numbers of curated thesauri, natural language documents, and database entries into a single entity for efficient access. BioQA solves the knowledge transformation challenge using a variety of custom algorithms for question analysis, concept and text snippet retrieval, and answer synthesis (automated summarization and paraphrasing). In an effort to make BioQA accessible and transparent we have also made all of the data in BioKB publicly available and have built a public web interface to serve the general public. The BioQA framework follows a standard Model-View-Controller (MVC) design, with BioQA’s web interface (the view), its underlying knowledge base BioKB (the model), and the collection of algorithms (the controller) being fully integrated to realize BioQA’s question answering capabilities. BioQA serves as useful framework to illustrate the potential of applying question answering, information retrieval and natural language processing techniques to the field of biomedicine. It also serves as a useful, publicly assessable web-based server to help researchers, educators and the general public address their information needs. We evaluated BioQA’s performance by participating in two separate BioASQ biomedical question answering challenges. BioQA performed exceptionally well and appears to be the top performing system. However, the results also make it clear that there is still room for future improvements for both BioQA and other QA systems. Given the progress to date and the growing utility that question-answering systems are already having, we expect many more high-performing, open access and domain-specific question answering systems will soon appear. These QA systems could have a significant and positive effect on how data is stored, how information is gained and how knowledge is acquired.

5. BioQA's Algorithmic Framework

BioQA uses a collection of algorithms to analyze user queries, perform concept and text snippet retrieval, transform document and concept retrieval results, and synthesize or paraphrase answers. Figure 27 shows an overview of BioQA's algorithmic framework and the relationships among its constituent modules.

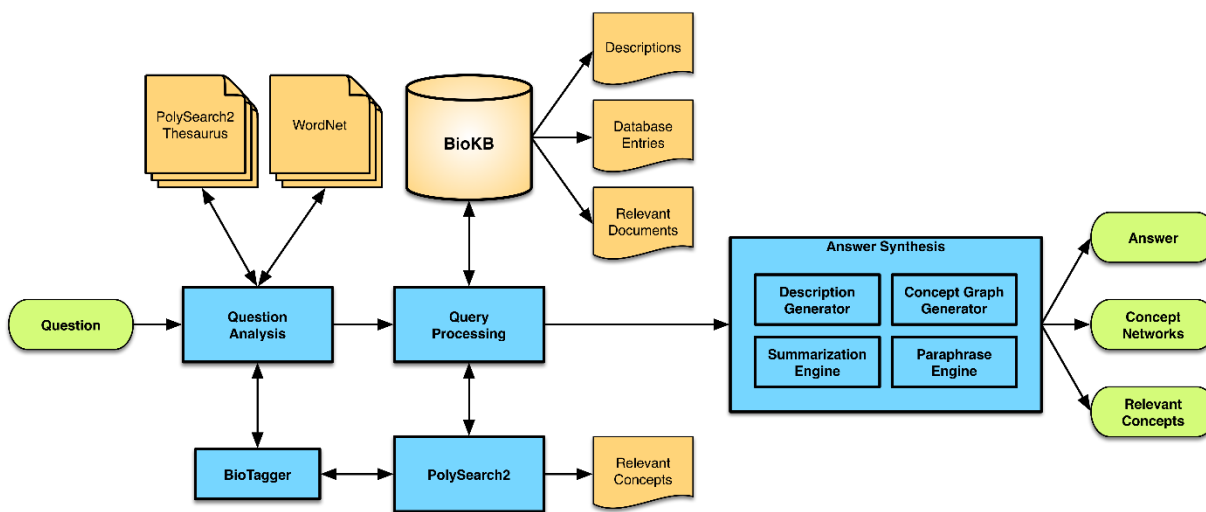


Figure 27: A flow chart showing BioQA's algorithms and the data flow through the system.

Given a question in natural language text, BioQA's "Question Analysis" module analyzes the question to extract question types, lexical answer types, query keywords, association words, and contextual noun phrases. Contextual noun phrases are noun phrases that are not query keywords but can be used to enhance search query formation. The Query Processing module formulates queries to search BioKB and PolySearch2. BioQA's Query Processing module retrieves key concepts in the question from BioKB's underlying ElasticSearch [71] index associated with BioKB and generates descriptions for each available concept using the "Description Generator". BioKB also contains concept networks spanned by relevant concepts, and houses co-mentioned concept networks from relevant documents. BioQA uses PolySearch2 [52] to retrieve relevant documents and snippets from an in-house document collection. PolySearch2 accepts a formulated search query and returns a list of relevant concepts and

snippets using the PolySearch2 algorithm [16, 17]. Based on the query analysis and query processing results, the “Answer Synthesis” module ranks relevant concepts, formats concept networks, and synthesizes textual answers. The final textual answers are synthesized using descriptions retrieved from the Description Generator. Summarization can also be done on relevant text snippets using BioQA’s greedy Latent Semantic Index (LSI) [56] based summarization algorithm, or summarization can also be done based on relationships among concepts in knowledge graphs. Users may also access the “Paraphrase module” which can be called to transform the final textual answer into a paraphrased paragraph with random syntactic variance.

In this chapter, I describe BioQA’s various algorithms for question analysis, named entity recognition, concept and snippet retrieval, description generation, answer synthesis, and automated paraphrasing.

5.1 Named Entity Recognition

Named Entity Recognition (NER) is a task for identifying concepts mentioned in a given text paragraph. These concepts could be implicitly expressed in various manifestations in the surface text (expressions that are actually used in a sentence). Named Entity Recognition require the parsing of surface text tokens corresponding to a certain concept, or Named Entity (NE). BioQA’s Named Entity Recognition (NER) module, called “BioTagger”, recognizes or “tags” biomedical terms mentioned in natural language text. Moreover, given a natural language sentence, BioTagger assigns words in the sentence as biomedical terms, query terms, association words, stop words, negation words, punctuation words, or non-keywords. BioTagger is an essential building block which serves multiple purposes within BioQA. These include: 1) extracting keywords from user question to form search queries in question analysis; 2) recognizing concepts mentioned in relevant snippets during concepts and snippets retrieval; 3) indexing relevant sentences by concepts for building co-mentioned networks and synthesizing answers. Figure 28 shows an example MEDLINE abstract tagged using BioTagger. Surface text tokens recognized as biomedical entities are tagged, color-coded, and hyperlinked to corresponding database records.

(2014) Hispidin derived from *Phellinus linteus* affords protection against acrylamide-induced oxidative stress in Caco-2 cells. *Chemico-biological interactions*; *Chem. Biol. Interact.*; 2014 Aug; 219:83-9

acrylamide (AA), a well-known toxicant, has attracted numerous attentions for its presumably carcinogenesis, neurotoxicity and cytotoxicity. Oxidative stress was considered to be associated with acrylamide cytotoxicity, but the link between oxidative stress and acrylamide cytotoxicity is still unclear. In the present study, hispidin produced from the edible fungus *Phellinus linteus* displayed dramatically antioxidant activities against DPPH radicals, ABTS radicals, ferric reducing and hydroxyl radicals, as well as superoxide anion radicals. Moreover, the cytoprotective effect of hispidin against AA-induced oxidative stress was verified upon Caco-2 cells according to evaluate the cell viability, intracellular ROS, mitochondrial membrane potential (MMP) and glutathione (GSH) in the presence or absence of AA (5 mM) in a dose-dependent manner. Collectively, our results demonstrated for the first time that hispidin was able to inhibit AA-induced oxidative stress, which might have implication for the dietary preventive application.

MEDLINE 24877638 : Hispidin derived from *Phellinus linteus* affords protection against acrylamide-induced oxidative stress in Caco-2 cells.

Figure 28: An example MEDLINE abstract tagged by BioTagger. Surface text tokens recognized as biomedical entities are tagged, color coded, and hyperlinked to corresponding database records.

The BioTagger algorithm combines exact dictionary matching against the BioKB thesauri, with noun phrase extraction, and N-gram language models. In the preprocessing stage, BioTagger tags stop words, association words, and punctuation using exact dictionary matches against a predefined list of such terms. In the term recognition stage, BioTagger first tries to recognize an exact match of any surface form of a biomedical concept; when no exact match is available, BioTagger uses Part-of-Speech (POS) tagging [46], Probabilistic Context-Free Grammar (PCFG) patterns [46], and regular expression patterns to extract noun phrases (NPs) as keywords; if no noun phrases are found, BioTagger generates frequent N-grams (for N ranging from 1 to 5) from the given sentences according to BioKB's MEDLINE N-gram dataset (available on the BioQA web server). In the above data processing steps, BioTagger prefers terms recognized using exact dictionary matches over noun phrases, or frequent N-grams. BioTagger also greedily prefers longer terms than shorter terms, as well as more frequent terms over less frequent terms. With algorithmic and implementation improvements, BioTagger is efficient in processing natural language sentences. Its memory efficient is linear to the size of thesauri, and its time efficiency is (best case) linear $O(N)$ or (worst case) $O(N^2)$ to length N of input sentence.

5.2 Question Analysis

BioQA's question analysis module extracts useful information from posted questions for downstream question answering process. Given a question, this module 1) predicts question types, 2) extracts Lexical Answer Types (LAT) [28], 3) extracts keywords, 4) extracts association words, and 5) extracts contextual noun phrases. Such information extracted from the given question is used to build search queries for concept and text snippet retrieval as well as answer synthesis (described in further details below).

BioQA supports both descriptive and associative question types. The question analysis module needs to determine whether the posted question is asking for a description of certain biomedical entities (descriptive) or finding associations between entities (associative). "Yes and no" questions are a special case of associative questions in the sense that such questions are looking at verifying the association as positive or negative between entities. Therefore, Yes/No questions are treated as associative questions. The question analysis module uses a rule-based algorithm to determine whether a posted question is descriptive or associative. This rule-based question type analysis algorithm examines the first five words of a posted question and predicts question types based on hand-crafted empirical rules. Here are a few examples in the collection of empirical rules on question prefix analysis:

- if a question starts with "which", "list", "name", "where", it is more likely an associative question;
- if a question starts with "can", "could", "has", "is", "are", "was", "were", "have", "does", "did", "should", etc., such question is more likely a yes/no question and therefore also an associative question in general.
- if a question starts with "describe", "what", "how", "define", "show", "explain", "provide", "elaborate", "who", or other verbs signifying actions or request, it is more likely a descriptive question.

We conducted an evaluation using the BioASQ's training dataset [88] of 100 questions, and BioQA's question type predictor achieve F-measure of 0.8026 for associative questions (0.9605

for yes/no questions), and 0.6936 for descriptive questions. Details for BioQA's question type analysis evaluation are presented in Chapter 4.

Besides determining the question type from a posted question, BioQA also analyzes the given question syntactically to identify query keywords, contextual noun phrases, lexical answer types, and association words. It is important for BioQA to recognize the main verb, subject, and predicate in a given sentence. BioQA first identifies the main verb in a given question through Part-of-Speech tagging, and uses shallow syntactic parsing to identify boundaries for the Subject and Predicate. Noun phrases (NPs) in a posted question are also important as they provide clues (query keyword or contextual NPs) for BioQA to search and filter for relevant document and snippets. Noun phrases are extracted from a question using regular expression pattern matching on a POS-tagged question. A sequence of words is defined as a noun phrase if they are:

- 1) one or more proper nouns,
- 2) one or more common nouns in singular or plural forms, or
- 3) a proper noun or prepositional phrase, followed by an optional adjective, followed by one or more common nouns.

NPs that are adjacent to the main verb in the predicate are treated as query keyword, while remaining NPs in the predicate are treated as contextual NPs. The query keyword is used for searching document collections for relevant documents, while contextual NPs are useful for ranking documents and filtering out relevant text snippets. Both query keyword and contextual NPs are used to formulate a customized PolySearch2 query for retrieving relevant concepts and snippets. Lexical Answer Types (LAT) are the type of the intended answer. For example, in a PolySearch2 query, the LAT is the type of biomedical entity or entities we wish to find. Consider the following a few examples (with the LAT underlined): “Which parasite causes malaria?”, “What diseases are associated with chemical BPA?”. BioQA uses a rule-based method to identify LATs in a posted question. BioQA extracts noun phrases in the subject between the query prefix words and the main verb of a sentence. It also uses predefined rules to map noun phrases to target LATs. Finally, verbs, adjectives, adverbs, and prepositions in the posted question are classified as association words.

5.3 Concepts and Snippets Retrieval

BioQA retrieves relevant concepts, documents, and snippets through PolySearch2. For more information on PolySearch2 [52], please refer to Chapter 3. Here we discuss how BioQA uses PolySearch2 to achieve its text mining objectives. From Question Analysis, BioQA forms a customized PolySearch2 query using information extracted from the input question and submits it to PolySearch2 for processing. PolySearch2 processes the customized query differently from other general queries, especially with regard to the following aspects: PolySearch2 uses the query keyword as a search term to find relevant documents, and then uses the lexical answer type and contextual noun phrases in ranking and filtering relevant text snippets. PolySearch2 then recognizes concepts (of all categories) mentioned in relevant snippets and scores them for relevant concepts. If too few concepts are recognized after filtering relevant concepts with an empirical cut-off, PolySearch2 performs an expanded query to include other noun phrases in the customized query.

BioQA uses PolySearch2 results in the following areas: 1) building co-mentioned concept networks from the list of relevant concepts and snippets, 2) synthesizing textual answers by combining PolySearch2's snippet results with BioKB's descriptions and relevant snippets in the summarization module. 3) integrating PolySearch2's result as relevant concepts for the input question.

Template Sentence Group	Example Generated Sentence
<ul style="list-style-type: none"> - DRUG_NAME is also known as DRUG_SYNONYM_LIST. - DRUG_NAME, also known as DRUG_SYNONYM_LIST, is a DRUG_CATEGORY. -..... 	<p>Moricizine, also known as Moracizinum, Ethmozin, Etmozin, Moracizine, or Moracizina, is an anti-arrhythmia agents and voltage-gated sodium channel blockers.</p>
<ul style="list-style-type: none"> - DRUG_NAME is used in the treatment of DISEASE_AND_CONDITION. - DRUG_NAME is used in the treatment of DISEASE_AND_CONDITION. INDICATION_SYNOPSIS. - DRUG_NAME is for used in treating DISEASE_AND_CONDITION. - 	<p>Moricizine is used used to treat irregular heartbeats (arrhythmias) and maintain a normal heart rate.</p>
<ul style="list-style-type: none"> - DRUG_NAME is branded as BRAND_NAME_LIST. - Major brands of DRUG_NAME are BRAND_NAME_LIST. - Known brands of DRUG_NAME are BRAND_NAME_LIST. - 	<p>Major brands of Moricizine are Ethmozine and Etmozins.</p>
<ul style="list-style-type: none"> - DRUG_NAME is a PHYSICAL_STATE. - DRUG_NAME is a PHYSICAL_STATE with a melting point of MELTING_POINT and boiling point of BOILING_POINT. - 	<p>This substance is a solid.</p>
<ul style="list-style-type: none"> - DRUG_NAME belongs to the CHEMICAL_CLASS_LIST group of drugs, which are known to act via the mechanism of action that MECHANISM_OF_ACTION_SYNOPSIS. - DRUG_NAME belongs to such chemical classes as CHEMICAL_CLASS_LIST. DRUG_CLASS_DESCRIPTION. - 	<p>This compound belongs to the phenothiazines.</p>

Table 19: Example sentence templates in a group and generated description for a DrugBank entry DB00680 Moricizine.

5.4 Description Generator

BioQA ensures each database entry in BioKB contains a description. BioQA extracts available description from database entries, and generates descriptions using description templates for database entries that are missing a description field, or when the original description is too short (e.g. less than 200 characters). Description Generator first parses a given database entry for target information fields and stores the extracted fields in a dictionary. It then generates descriptions by filling in corresponding blanks using pre-defined sentence templates and produces a description paragraph. Description templates consist of sentence templates organized in multiple sentence groups. Appendix A shows some example description templates for DrugBank [8] entries. Table 19 shows an example sentence group and examples of generated descriptive sentences.

A sentence group represents a single sentence describing one or more properties for a database entry. Each sentence group contains multiple hand-crafted sentence templates conveying similar information in different syntactic variations. Each sentence template contains one or more blank fields (uppercase words as shown in Table 19) to be filled with information extracted from the corresponding database entry. A sentence template is “triggered” if all blank fields have the corresponding information extracted from the database entry. It is common that a database entry simultaneously triggers multiple templates in a sentence group. In that case, templates with a greater number of satisfied blank fields (hence carrying more information) are preferred. Finally, one triggered template in the group is selected at random to produce a descriptive sentence to induce artificial syntactic variations to give the impression of human editing. If no templates are triggered due to missing information, no sentence is produced for such a sentence group. Sentence templates shown in Table 19 contain multiple sentence templates describing a drug chemical’s name, physical state, melting and boiling point. If a DrugBank entry contains drug name, physical state, and melting point, any sentence template containing all or some of these fields are “triggered”. The Description Generator then randomly selects a sentence template to fill in blank fields and produce the final descriptive sentence. Description Generator iterates through each sentence group to produce descriptive sentences, and organizes them into a descriptive paragraph based on the pre-defined order of sentence groups. Sentences are organized into a paragraph in a fixed order to improve readability. For example,

the sentence groups in Appendix A describe a DrugBank drug chemical by first describing its name, synonyms, and drug category. Then the description provides information about diseases or conditions that the drug is intended to treat and the indications for the medication; the text then describes brands and manufacturers for the drug and drug approval information. It further discusses the drug's physical state, melting point and boiling point (shown in Table 19), chemical and drug class, mechanism of action, absorption, half-life, and route of elimination. Finally, the description paragraph describes interacting drugs, drug targets, and catalyzing enzymes. Appendix A shows more examples of several generated descriptions with corresponding original DrugBank descriptions for comparison. Generated descriptions, along with extracted descriptions from database entries, are indexed in BioKB.

5.5 Answer Synthesis

BioQA synthesizes answers in natural language using automated summarization. First of all, BioQA retrieves descriptions from BioKB's description collection and composes a descriptive paragraph to describe key concepts identified in the input question. Besides this first descriptive paragraph, BioQA synthesizes further answer paragraphs by summarizing sentences describing concept associations in a co-mentioned concept network, and a document index built from relevant concepts and snippets retrieved from the PolySearch2 results. This section discusses BioQA's summarization algorithm in using both data structures.

BioQA summarizes answer paragraphs describing associations between relevant concepts using a co-mentioned concept graph, built on-the-fly from relevant concepts and snippets. Nodes in the "concept graph" are biomedical entities, and edges in the graph indicate associations between entities. Nodes can be biomedical entities identified from the input question (query nodes), or from retrieved relevant snippets (relevant nodes). Edges represent an association between two connecting nodes. Edges are non-directional and are weighted based on the co-occurrence frequency of entities represented by the two connecting nodes. BioQA builds a concept graph on-the-fly from relevant snippets retrieved for a given question. BioQA keeps track of the co-occurrence frequency for each pair of concepts mentioned in the relevant sentences. Concept pairs with co-occurrence frequencies above an empirical threshold establish

an edge in the concept graph, with relevant sentences mentioning such concept pairs as supporting evidence. BioQA then transforms the concept graph to a summary paragraph. BioQA first identifies the target subgraph spanned by all question nodes. The target subgraph consists of all question nodes, plus any relevant nodes along the shortest paths between each pair of question nodes. This target subgraph represents the “concept space” spanned by a given question. BioQA then traverses the target subgraph, along shortest paths between each pair of target nodes, and joins the supporting sentences to form a summary paragraph in the order of traversal. If there is more than one supporting sentence along an edge, a single supporting sentence is selected at random among top ranked (according to PolySearch2’s relevancy scores) supporting sentences. Figure 29 shows a pseudocode algorithm for BioQA’s summarization algorithm with a co-mentioned concept graph. Figure 30 shows pseudocode for building a concept graph.

Algorithm 1 BioQA-GraphSum: Summarization by Co-occurrence Concept Graph

Require: Q : a question sentence

Require: R : an array of sentences (relevant snippets)

```

1: return  $P$ : an ordered array of sentences (summary paragraph)
2:  $P \leftarrow$  sentence array
3: {Build co-occurrence concept graph  $G$  using Algorithm Build-Concept-Graph}
4:  $G \leftarrow$  Build-Concept-Graph( $Q, R$ )
5: Extract target nodes  $ts$  from question sentence  $Q$ 
6: for all node pair  $(t1, t2)$  in target nodes  $ts$  do
7:   {find all nodes along the shortest path between target nodes  $t1$  and  $t2$ }
8:    $shortest - path \leftarrow$  Find-Shortest-Path( $G, t1, t2$ )
9:   if  $shortest - path$  is not empty then
10:    for all node pair  $(sp1, sp2)$  in shortest-path do
11:      {retrieve a representative sentence}
12:      {from all relevant sentences mentioning both concepts  $sp1$  and  $sp2$ .}
13:       $s \leftarrow$  Find-Representative-Support-Sentence( $R, sp1, sp2$ )
14:       $P \leftarrow P + s$ 
15:    end for
16:  end if
17: end for
18: return  $P$ 

```

Figure 29: BioQA's algorithm on summarization via the co-occurrence concept graph.

Algorithm 2 Build-Concept-Graph: build concept graph from relevant snippets

Require: Q : a question sentence

Require: R : an array of sentences (relevant snippets)

```
1: return  $G$ : a co-occurrence concept graph
2: Initialize dictionary  $d$  to count frequency of node pairs
3: {Initialize node set  $nodes$  and edge set  $edges$  for graph  $G$ }
4:  $nodes \leftarrow \emptyset$ 
5:  $edges \leftarrow \emptyset$ 
6: {Extract target nodes  $ts$  from question sentence  $Q$ }
7:  $ts \leftarrow \text{Named-Entity-Recognizer}(Q)$ 
8:  $nodes \leftarrow nodes \cup ts$ 
9: for all  $sentence$  in  $R$  do
10:   {Extract relevant nodes  $ns$  from  $sentence$ }
11:    $sentence \leftarrow \text{Named-Entity-Recognizer}(sentence)$ 
12:    $nodes \leftarrow nodes \cup ns$ 
13:   for all node pair  $(n1, n2)$  in relevant nodes  $ns$  do
14:      $d[n1, n2] \leftarrow d[n1, n2] + 1$ 
15:   end for
16: end for
17: {Calculate frequency threshold  $f^*$  with values in  $d$ }
18:  $f^* \leftarrow \text{Calculate-Frequency-Threshold}(d)$ 
19: for all node pair  $(n1, n2)$  in  $d$  do
20:   if  $d[n1, n2] \leq f^*$  then
21:      $edges \leftarrow edges \cup (n1, n2)$ 
22:   end if
23: end for
24:  $G \leftarrow (nodes, edges)$ 
25: return  $G$ 
```

Figure 30: The Build-Concept-Graph algorithm builds concept graphs from relevant text snippets.

Algorithm 3 BioQA-GreedyLSI: Greedy summarization using document matrix and Latent Semantic Indexing

Require: Q : a question sentence

Require: R : an array of sentences (relevant snippets)

Require: l : summary length limit (an integer)

```
1: return  $P$ : an ordered array of sentences (summary paragraph)
2: Initialize sentence array  $P$ 
3: {calculate term frequency statistics for both question  $Q$  and relevant snippets  $R$ }
4:  $d \leftarrow$  Calculate-Term-Frequency ( $Q, R$ )
5: {build vector space model using latent semantic indexing}
6:  $M \leftarrow$  Build-Vector-Space-Model ( $Q, R, d$ )
7:  $qt \leftarrow$  Convert-Term-Array ( $d, Q$ ) {convert question  $Q$  to a term array}
8:  $P \leftarrow P + qt$  {add question to list of sentence in the summary}
9: {find representative support sentence from vector space model  $M$ }
10:  $s \leftarrow$  Find-Sentence-With-Highest-Cosine-Similarity ( $M, P$ )
11:  $P \leftarrow P + s$ 
12: {find and remove sentences very similar to the current support sentence}
13:  $ss \leftarrow$  Find-Sentences-With-High-Cosine-Similarity ( $M, s$ )
14:  $R' \leftarrow$  Remove-Sentences ( $R, ss$ )
15: while (Length ( $P$ )  $\geq l$ ) and ( $R'$  is not empty) do
16:   {update vector space model}
17:    $M' \leftarrow$  Build-Vector-Space-Model ( $Q, R', d$ )
18:   {find representative support sentence from updated vector space model  $M$ }
19:    $s \leftarrow$  Find-Sentence-With-Highest-Cosine-Similarity ( $M', P$ )
20:    $P \leftarrow P + s$ 
21:   {find and remove sentences very similar to the current support sentence}
22:    $ss \leftarrow$  Find-Sentences-With-High-Cosine-Similarity ( $M', s$ )
23:    $R' \leftarrow$  Remove-Sentences ( $R', ss$ )
24: end while
25:  $P \leftarrow P - qt$  {remove question sentence from summary}
26: {retrieve original sentences in  $R$  to replace term vectors in  $P$ }
27:  $P \leftarrow$  GET-ORIGINAL-SENTENCES ( $P, R$ )
28: return  $P$ 
```

Figure 31: BioQA's summarization algorithm using document matrix and Latent Semantic Indexing techniques.

Algorithm 4 Build-Vector-Space-Model: build and update vector space models from a collection of relevant snippets

Require: Q : a question sentence

Require: R : an array of sentences (relevant snippets)

Require: d : term frequency dictionary

- 1: **return** M : a matrix representation of vector space model built from R
- 2: Initialize matrix data structure M
- 3: Initialize Rt , an array of term arrays
- 4: $qt \leftarrow$ **Convert-Term-Array** (d, Q) {convert question Q to a term array}
- 5: $Rt \leftarrow Rt + qt$ {add question term array to list of term arrays}
- 6: **for all** relevant sentence r in array R **do**
- 7: $rt \leftarrow$ **Convert-Term-Array** (d, r) {convert sentence r to a term array}
- 8: $Rt \leftarrow Rt + rt$ {add term array to list of term arrays}
- 9: **end for**
- 10: {calculate eigenvectors V of matrix Rt (array of term arrays)}
- 11: $V \leftarrow$ **Latent-Semantic-Indexing** (Rt)
- 12: $M \leftarrow$ **Convert-Vector-Space-Model** (Rt, V)
- 13: **return** M

Figure 32: BioQA's automatic summarization algorithm for building a vector space model from retrieved text snippets.

BioQA also generates a summary paragraph from relevant text snippets using Latent Semantic Indexing (LSI) [56] on a document matrix. This additional summary is needed when a target concept graph is not available or the summary generated from the concept graph is too short. Figure 31 shows the pseudocode algorithm for generating descriptions using the document matrix and the greedy Latent Semantic Index. Figure 32 shows the pseudocode algorithm for building document matrix from a collection of relevant documents. BioQA first builds an LSI data structure from relevant concepts (Figure 32), then greedily retrieves the next most similar sentence to the current summary, and then updates both the current summary and the LSI data structure until the summary reaches a certain length or the LSI data structure is exhausted. Given a question and a collection of relevant snippets. BioQA converts relevant snippets to a document vectors and form a document matrix using a vector space model. Rows in the document matrix represent text snippets, while columns in the document matrix represent terms (or topics). BioQA then calculates eigenvalues and eigenvectors of the document matrix using Singular

Value Decomposition (SVD) and projects the document matrix to a lower dimension. This step effectively filters key terms (topics) among the collection of relevant snippets and indexes each snippet with key terms. Next, BioQA greedily forms a summary paragraph using the initial question document vector and the document index in subsequent iterations (Figure 31). Given an initial question document vector, BioQA retrieves snippets corresponding to the most similar document vector in the document index by Cosine Similarity. BioQA adds the retrieved snippets to the summary paragraph, removes snippets similar to the current snippet above an empirical threshold, and recalculates the document index, now containing fewer documents. This process is repeated until the summary paragraph grows to a certain length, or the document matrix contains too few relevant snippets to continue the indexing process.

BioQA generates summary paragraphs using both of the above algorithms and also performs post-processing to produce final summary paragraphs. During the post-processing step, BioQA rearranges sentences within summary paragraph to enhance readability and fixes grammatical artifacts introduced during the summarization process. Synthesizing answer paragraphs using the document matrix with latent semantic indexing is a process of information filtering for identifying key terms and key snippets among all relevant snippets. On the other hand, synthesizing answers using a concept graph built on co-mentioned concepts represents a process of implicit reasoning, where we join sentences describing connections of entity across multiple connections in the natural order that are found to be associative in relevant snippets.

5.6 Paraphrasing Module

In addition to synthesizing textual answers, BioQA also provides an optional paraphrasing function for users to include all or part of the synthesized answer in their own work without the need to manually paraphrase BioQA's synthesized answer. BioQA's paraphrasing module accepts an initially synthesized textual answer, and paraphrases it, sentence by sentence, according to a set of substitution, enumeration, rearrangement, and transformation rules. Please refer to Appendix B for examples of rules corresponding to these categories. This section briefly describes each category of rules and their applications in automated paraphrasing.

The paraphrase engine applies phrase substitution, word-sense substitution, and synonym substitution to an input sentence. The paraphrase engine first applies 2000+ phrase or word substitution rules (see Appendix B) to an input sentence to replace a phrase with its semantic equivalent. These substitution rules can be simple or word-sense dependent (substitution rules depends on the Part-of-Speech tag for the original words). Simple substitution replaces a phrase with an equivalent phrase. For example, substituting "also known as" with "also referred to as". Word-sense substitution substitutes a word based on its Part-of-Speech tag. For example, the word "witness" can be substitute with "observe" when "witness" is used as a verb, but with "observer" when "witness" is used as a noun. The paraphrase engine then substitutes a word with a valid synonym by searching WordNet [62] (English dictionary words) and the PolySearch2 biomedical thesaurus [52] (biomedical terms). The paraphrase engine recognizes phrases or common expressions such that synonyms substitution does not replace part of a phrase or common expression by mistake. Besides substitutions, the paraphrase engine also performs transformation, enumeration, and rearrangement rules to paraphrase an input sentence. Transformation rules changes a numerical measure to an equivalent expression with different units. For example, changing "1000 feet" to "305 meters", or "10 lb" to 4.5 kg". If the input sentence contains an enumeration of several items, the paraphrase engine randomizes the order of items within the enumeration. For example, changing "animals, plants, and fungi" to "plants, fungi, and animals". The paraphrase engine also rearranges words in an expression. For example, changing "The A of the B" to "The B's A", and changing "A said B" to "B, said A". These rules further paraphrase an input sentence after substitutions.

In paraphrasing, the module also obeys other rules that don't fit into the previous categories. For example, it should never change anything in quotes, and never change proper nouns, acronyms ("BPA") or entity names ("Bisphenol A"). This paraphrasing process preserves quotes by first extracting quotes from the sentence, paraphrasing the rest, and re-substituting the quote back to the paraphrased results. The paraphrase module also detects acronyms (spelled entirely in uppercase letters), proper nouns (through syntactic parsing) and entity names (through the use of our biomedical thesauri), and preserves these during paraphrasing. Finally, the paraphrasing module goes over sentences to detect and fix any article errors (e.g. Changing "a immature" to "an immature", and changing "an historical" to "a historical") introduced accidentally during paraphrasing. When multiple rules are applicable to an input sentence, there could be a potential conflict between rules, as more than one rule could be substituting the same part of the sentence yielding different results. In this case, only one rule is selected among the conflicting rules (according to rule precedence or at random) to paraphrase a sentence. Besides conflicting rules, BioQA also randomizes paraphrasing results to a certain degree to provide higher degree of syntactic variance. Running the paraphrasing function again will yield a slightly different result based on the same synthesized answer.

5.7 Conclusion

In this chapter, I described BioQA's algorithmic framework for named entity recognition, question analysis, concepts and snippets retrieval, description generation, answer synthesis, and automated paraphrasing. These algorithms are crucial components of BioQA. Working together as a whole, these algorithms transform input question to search queries for finding relevant concepts and snippets, and then further derive an answer summary from the retrieved documents and database records.

6. Concluding Remarks

The central objective of this thesis was to advance the field of biomedical question answering. Traditionally QA systems have focused on answering simple questions or general knowledge questions in the open-domain. These might include “*What is the temperature in Edmonton today?*” or “*What is the population of Canada?*” Recently there have been significant advances in the open domain for more difficult tasks associated with question answering, particularly with the roll-out of IBM's Watson [28] on *Jeopardy!*. However, the field of biomedicine still attracts relatively little attention with regard to question answering.

In this thesis, I hypothesized that with existing technology, it would be possible to build a prototype biomedical QA system that could significantly advance the field of QA in biomedicine. This served as the motivation to design and implement a comprehensive, end-to-end QA system called BioQA for biomedical question answering. Noting that a high throughput search engine is crucial for BioQA, I started my research by building around the PolySearch algorithm [16, 17]. PolySearch was previously developed in 2006-2008 to perform targeted text mining of the PubMed/Medline text corpus. I expanded PolySearch to be a much more general purpose biomedical search engine (PolySearch2) [52] that could be used to retrieve relevant concepts and text snippets from a far wider variety of databases. I also spent much effort in creating and curating a controlled vocabulary and dictionary (i.e. the PolySearch2 thesaurus collection) as well as maintaining a much larger and more comprehensive collection of databases and text resources. I also developed a number of speed-ups, hardware modifications and algorithmic improvements that reduced search times in PolySearch from 10s of minutes to mere seconds. I further demonstrated that the new version of PolySearch was able to out-perform the old version of PolySearch in many different search and query tasks (see Chapter 3 for more details). With the completion of PolySearch2 and the assembly of some of the key data infrastructure, I began to explore the next phase of the biomedical QA challenge. In particular, I proposed and implemented a number of innovative algorithms (Chapter 5) to transform input questions into search queries, to perform accurate ranking of relevant text snippets, to synthesize and paraphrase natural language answers, as well as to generate informative concept graphs. I evaluated BioQA's performance for its various modules using local experiments as well as on

the shared tasks of BioASQ challenge [88, 89]. The results from this open and objective assessment showed that BioQA performs significantly better than all current biomedical QA systems. In an effort to make this resource open and accessible to all, I built a user interface, the BioQA web server, and implemented a number of useful graphical displays and interactive functions (Chapter 4).

Through the implementation of BioQA, I learned that 1) a comprehensive biomedical thesaurus is essential for almost all steps of biomedical question answering, and 2) effective summarization algorithms are key to deriving natural language answers from relevant concepts and snippets. A comprehensive biomedical thesaurus is crucial for query processing (parsing named entities from input question), documents and snippets retrieval (indexing and retrieving documents based on biomedical concepts), and answer synthesis (weighting and organizing sentences in summary using concepts mentioned in relevant snippets). A collection of effective summarization algorithms, either using statistical summarizations or paths in a concept graph, to join relevant sentences and form natural language answers is essential to convert a seemingly random collection of relevant sentences and snippets to form a comprehensive summary. Therefore, enrichment of BioQA's biomedical thesaurus and enhancements to BioQA's summarization algorithms should effectively boost BioQA's overall performance.

While the current implementation of BioQA offers many positive and useful features, there are a number of capabilities or features that could be added to make it better. Currently, BioQA is a state-less QA or state-less query system, meaning that it treats each question or query as a brand new query and does not make reference to previous searches. I believe it could be possible to enhance BioQA to be an "adaptive QA" system that constantly improves its answers based on previous query submissions. In order to adapt to an individual, BioQA could be modified to automatically build a custom collection of search keywords from a user's previous searches, and to use this keyword list to adjust its ranking scores and the way it performs its answer synthesis. By providing an interactive tool (through BioQA's web site) it may be possible for users to rank returned answers thereby acquiring knowledge or training data for BioQA to adapt to a user's specific needs. By keeping track of a series of questions asked by the same user and taking user feedback into consideration for subsequent searches, BioQA could progressively improve upon itself and adapt to individual user needs. BioQA currently lacks the capacity to

perform logical or semantic reasoning. Modern QA systems are moving towards reasoning and semantic processing to enhance their question answering capabilities and user experience. For example, IBM Watson supports a certain degree of semantic reasoning through the use of semantic “frames” that encapsulate semantic relations. Knowledge Engines such as Wolfram Alpha support certain logical reasoning operations like solving simple mathematical equations. In order for BioQA to support reasoning, it needs to convert user questions to more than just search queries, but also to logical representation to validate against a collection of logical entailments representing existing biomedical knowledge. This will represent a significant challenge, but as more biomedical databases become more logically structured, this may soon be possible. I also believe that BioQA could be designed to take advantage of domain knowledge available within a smaller subfield. For example, within certain subfields, highly structured or curated information exists. For example, SMPDB [45] captures knowledge on biochemical pathways and reactions between chemicals and enzymes. Therefore, SMPDB can be used to make certain inferences on biochemical pathways. In this regard it may be possible to build and attach specific “inference engines” with subfield-specific knowledge to BioQA, thereby enhancing its question answering capability with basic inference.

This work represents one of the first efforts to bring QA concepts into the biomedical domain. I believe BioQA is the first biomedical QA system to integrate such a broad range of databases and offer such a broad range of capabilities. It is also the first QA system capable of searching both highly structured databases and natural language text databases. I believe BioQA represents an important new step in the development of text retrieval and data mining tools for biomedical research. Continuing our research on the BioQA framework proposed here, we could transform not only the way researchers, physicians, educators and the general public use the web but also how they learn and do their research.

Bibliography

- [1] Akella, L.M., Norton, C.N., Miller, H. (2012) NetiNeti: discovery of scientific names from text using machine learning methods. *BMC Bioinformatics*. Aug 22;13:211.
- [2] Anwar, S. (2014). Representing, Reasoning and Answering Question about Biological Pathways Various Applications. Dissertation, Arizona State University, *Ann Arbor: ProQuest/UMI*, May.
- [3] Apache Software Foundation. (2016) ElasticSearch. <http://www.elastic.co> (Accessed July 11).
- [4] Arndt, D., Xia, J., Liu, Y., Zhou, Y., Guo, A.C., Cruz, J.A., Snelnikov, I., Budwill, K., Nesbo, C.L., Wishart, D.S. (2012) METAGENassist: a comprehensive web server for comparative metagenomics. *Nucleic Acids Research*. Jul;40(Web Server issue):W88-95.
- [5] Aronson, A.R., Lang, F.M. (2010) An overview of MetaMap: historical perspective and recent advances. *Journal of American Medical Informatics Association*. May-Jun;17(3):229-36.
- [6] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T. et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25, 25-29.
- [7] Athenikos, S.J., and Han, H. (2010) Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1-24, July.
- [8] Averill, R.F., Mullin, R.L., Steinbeck, B.A., Goldfield, N.I. and Grant, T.M. (1998) Development of the ICD-10 Procedure Coding System (ICD-10-PCS). *Journal of AHIMA / American Health Information Management Association*, 69, 65-72.
- [9] Baasiri, R.A., Glasser, S.R., Steffen, D.L. and Wheeler, D.A. (1999) The breast cancer gene database: a collaborative information resource. *Oncogene*, 18, 7958-7965.
- [10] Bagchi, S., Ferrucci, D.A., Gondek, D., Levas, A., and Mueller, E.T. (2013) Watson: Beyond jeopardy! *Artificial Intelligence*, 199:93-105. August.
- [11] Baker, C., Fillmore C., and Lowe J. (1998) The Berkeley FrameNet project. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL 1998, pages 86-90, Stroudsburg, PA, USA.

- [12] Balikas, G., Partalas, I., Ngomo, A.N., Kirthara, A., Gaussier, E., Paliouras, G. (2014) Result of the BioASQ Track of the Question Answering Lab at CLEF 2014. In *CLEF2014 Working Notes. Working Notes for CLEF 2014 Conference*. Sheffield, UK, September 15-18.
- [13] Bekhuis, T. (2006) Conceptual biology, hypothesis discovery, and text mining: Swanson's legacy. *Biomedical Digital Library*. Apr 3;3:2.
- [14] Bodenreider, O. (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*. Jan 1;32(Database issue):D267-70.
- [15] Caspi, R., Altman, T., Billington, R., Dreher, K., Foerster, H., Fulcher, C.A., Holland, T.A., Keseler, I.M., Kothari, A., Kubo, A. et al. (2014) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Research*, 42, D459-471.
- [16] Cheng, D., (2007) PolySearch: A Web based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. (Master's thesis). Retrieved from ProQuest Dissertations and Thesis. *Ann Arbor: ProQuest/UMI* (Association Order No. [MR33217])
- [17] Cheng, D., Knox, C., Young, N., Stothard, P., Damaraju, S. and Wishart, D.S. (2008) PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Research*, 36, W399-405.
- [18] Clark, K., Karsch-Mizrachi I., Lipman D.J., Ostell J., Sayers E.W. (2016) GenBank. *Nucleic Acids Research*. Jan 4;44(D1):D67-72.
- [19] Corlan, A.D. (2004) Medline trend: automated yearly statistics of PubMed results for any query. <http://dan.corlan.net/medline-trend.html> (Accessed February).
- [20] Cormen, T., Leiserson, C., Rivest, R., Stein, C. (2001) Introduction to Algorithm. (2nd ed.). *McGraw-Hill Higher Education*.
- [21] Cruz, J., Liu, Y., Liang, Y., Zhou, Y., Wilson, M., Dennis, J.J., Stothard, P., Van Domselaar G., Wishart, D.S. (2012) BacMap: an up-to-date electronic atlas of annotated bacterial genomes. *Nucleic Acids Research*. Jan;40(Database issue):D599-604.
- [22] Doms, A. and Schroeder, M. (2005) GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Research*, 33, W783-786.
- [23] Dong, X.L., Halevy, A.Y., Yu, C. (2009) Data integration with uncertainty. *The VLDB Journal*. Apr;18(2):469-500.

- [24] Fan, J., Kalyanpur, A., Gondek, D. C., and Ferrucci, D. A. (2012) Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56(3.4), 5:1–5:10.
- [25] Fan, J., Kalyanpur, A., Gondek, D. C., Ferrucci, D. A. Allen, P.G., Angele, J., Baxter, D., Barker, K., Curtis, J., Chaudhri, V.K., Chaw, S.Y., Clark, P., Friedland, N.S., Fan, J., Israel, D.J., Matthews, G., Miraglia, P., Mönch, E., Oppermann, H., Porter, B.W., Shepard, B., Staab, S., Tecuci, D., Witbrock, M.J., Wenke, D., & Yeh, P.Z. (2004). Project Halo: Towards a Digital Aristotle. *AI Magazine*, 25:29-48.
- [26] Faro, A., Giordano, D., Spampinato, C. (2012) Combining literature text mining with microarray data: advances for system biology modeling. *Briefings in Bioinformatics*. 13(1), 61-82.
- [27] Fernald, G.H., Capriotti, E., Daneshjou, R., Karczewski, K.J., Altman, R.B. (2011) Bioinformatics challenges for personalized medicine. *Bioinformatics*. 27(13), 1741-1748.
- [28] Ferrucci, D., Brown E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefter, N., and Welty, C. (2010) Building watson: an overview of the DeepQA project. *AI Magazine*, September.
- [29] Fontaine, J.F., Priller, F., Barbosa-Silva, A., Andrade-Navarro, M.A. (2011) Genie: literature-based gene prioritization at multi genomic scale. *Nuclear Acids Research*, 39 (Web Server issue), W455-461.
- [30] Forney, G.D. (1973) The viterbi algorithm. in *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278.
- [31] Gasteiger, E., Jung, E. and Bairoch, A. (2001) SWISS-PROT: connecting biomolecular knowledge via a protein database. *Current issues in molecular biology*, 3, 47-55.
- [32] Gerner, M., Nenadic, G., Bergman, C.M. (2010) LINNAEUS: a species name identification system for biomedical literature. *BMC Bioinformatics*. Feb 11;11:85.
- [33] Google (2006) Google N-Gram Data, <http://googleresearch.blogspot.ca/2006/08/all-our-n-gram-are-belong-to-you.html> (Accessed November).
- [34] Gu, B., Kashani, M.M., Liu, Y., Melli, G., Popowich, F., Shi, Z., Sarkar, A., & Wang, Y. (2007). Question Answering Summarization of Multiple Biomedical Documents. In *Proceedings of the 20th conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, 284-295, May 28-30.

- [35] Guo, A.C., Jewison, T., Wilson, M., Liu, Y., Knox, C., Djoumbou, Y., Lo, P., Mandal, R., Krishnamurthy, R. and Wishart, D.S. (2013) ECMDB: the E. coli Metabolome Database. *Nucleic Acids Research*, 41, D625-630.
- [36] Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A. and McKusick, V.A. (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33, D514-517.
- [37] Hastie, T., Tibshirani, R., and Friedman, J. (2003) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Springer*, corrected edition.
- [38] Hatcher, E., and Gospodnetic, O. (2004) Lucene in Action. *Manning Publications*.
- [39] Hettne, K.M., Stierum, R.H., Schuemie, M.J., Hendriksen, P.J., Schijvenaars, B.J., Mulligen, E.M., Kleinjans, J., Kors, J.A. (2009) A dictionary to identify small molecules and drugs in free text. *Bioinformatics*. 25(22):2983-91.
- [40] Hirschman, L., Burns, G.A., Krallinger, M., Arighi, C., Cohen, K.B., Valencia, A., Wu, C.H., Chatr-Aryamontri, A., Dowell, K.G., Huala, E., Lourenço, A., Nash, R., Veuthey, A.L., Wieggers, T., Winter, A.G. (2012) Text mining for the biocuration workflow. *Database (Oxford)*. Apr 18; 2012:bas020.
- [41] Hoy, M.B. (2010). Wolfphram Alpha: A brief introduction. *Medical Reference Services Quarterly*, 29(1):6774.
- [42] Humphreys, B.L., Lindberg, D.A., Schoolman, H.M. and Barnett, G.O. (1998) The Unified Medical Language System: an informatics research collaboration. *Journal of the American Medical Informatics Association: JAMIA*, 5, 1-11.
- [43] Hur, J., Schuyler, A.D., States, D.J., Feldman, E.L. (2009) SciMiner: web-based literature mining tool for target identification and functional enrichment analysis. *Bioinformatics*. 25(6), 838-840.
- [44] Jewison, T., Knox, C., Neveu, V., Djoumbou, Y., Guo, A.C., Lee, J., Liu, P., Mandal, R., Krishnamurthy, R., Snelnikov, I. et al. (2012) YMDB: the Yeast Metabolome Database. *Nucleic Acids Research*, 40, D815-820.
- [45] Jewison, T., Su, Y., Disfany, F.M., Liang, Y., Knox, C., Maciejewski, A., Poelzer, J., Huynh, J., Zhou, Y., Arndt, D., Djoumbou, Y., Liu, Y., Deng, L., Guo, A.C., Han, B., Pon, A., Wilson, M., Rafatnia, S., Liu, P., Wishart, D.S. (2014) SMPDB 2.0: big improvements to

the Small Molecule Pathway Database. *Nucleic Acids Research*. Jan;42(Database issue):D478-84.

- [46] Jurafsky D., and Martin J.H. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall Series in Artificial Intelligence*. Prentice Hall, 1 edition, February.
- [47] Kanehisa, M. and Goto, S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28, 27-30.
- [48] Kim, M.Y., Dou, Q., Zaiane, O.R., and Goebel, R. (2010) Unsupervised mapping of sentences to biomedical concepts based on integrated information retrieval model and clustering. *In Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology, BCB 10*, pages 322-329, New York, NY, USA.
- [49] Knox, C., Shrivastava, S., Stothard, P., Eisner, R., Wishart, D.S. (2007) BioSpider: a web server for automating metabolome annotations. *Pacific Symposium on Biocomputing*. 145-56.
- [50] Krallinger, M., Leitner, F., Vazquez, M., Salgado, D., Marcelle, C., Tyers, M., Valencia, A., Chatr-aryamontri, A. (2012) How to link ontologies and protein-protein interactions to literature: text-mining approaches and the BioCreative experience. *Database (Oxford)*. Mar 21;2012:bas017.
- [51] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A.C., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., Neveu, V. et al. (2014) DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, 42, D1091-1097.
- [52] Liu, Y., Liang, Y., Wishart, D. (2015) PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Research*. Jul 1;43(W1):W535-42.
- [53] Lu, C.J., Tormey, D., McCreedy, L., Browne, A.C., (2014) Using Element Words to Generate (Multi)Words for the SPECIALIST Lexicon. *AMIA 2014 Annual Symposium*, Washington., DC, Nov. 15-19, P.1499.
- [54] Lu, Z. (2011) PubMed and beyond: a survey of web tools for searching biomedical literature. *Database: the journal of biological databases and curation*, baq036.
- [55] Malakasiotis, P., I. Androutsopoulos, Y. Almirantis, D. Polychronopoulos, and I. Pavlopoulos, (2013) Tutorials and Guidelines. *BioASQ, Project Deliverable D3.4*, Jan.

- [56] Manning C.D. and Schuetze, H. (1999) Foundations of Statistical Natural Language Processing. *The MIT Press*, 1 edition, June.
- [57] Mao, Y., Wei, C.H., Lu, Z. (2014) NCBI at the 2014 BioASQ challenge task: large-scale biomedical semantic indexing and question answering. In: CLEF2014 Working Notes. *CLEF 2014 Workshop Proceedings*. Sheffield, UK. September 15-18.
- [58] Marcus, M.P., Marcinkiewicz, M.A., and Santorini, B. (1993) Building a large annotated corpus of English: the penn treebank. *Computational. Linguist.* 19:2, 313-330.
- [59] McEntyre, J.R., Ananiadou, S., Andrews, S., Black, W.J., Boulderstone, R., Buttery, P., Chaplin, D., Chevuru, S., Cobby, N., Coleman, L.A. et al. (2011) UKPMC: a full text article resource for the life sciences. *Nucleic Acids Research*, 39, D58-65.
- [60] Microsoft. (2009) Bing. <http://www.bing.com/> (accessed July).
- [61] Millar, J. (2016) The Need for a Global Language - SNOMED CT Introduction. *Studies in Health Technology and Informatics*. 225:683-5.
- [62] Miller G. (1995) WordNet: A Lexical Database for English. *Communications of the ACM* Vol. 38, No. 11:39-41.
- [63] Mishra G.R., Suresh M., Kumaran K., Kannabiran N., Suresh S., Bala P., Shivakumar K., Anuradha N., Reddy R., Raghavan T.M., Menon S., Hanumanthu G., Gupta M., Upendran S., Gupta S., Mahesh M., Jacob B., Mathew P., Chatterjee P., Arun K.S., Sharma S., Chandrika K.N., Deshpande N., Palvankar K., Raghavnath R., Krishnakanth R., Karathia H., Rekha B., Nayak R., Vishnupriya G., Kumar H.G., Nagini M., Kumar G.S., Jose R., Deepthi P., Mohan S.S., Gandhi T.K., Harsha H.C., Deshpande K.S., Sarker M., Prasad T.S., Pandey A. (2006) Human protein reference database--2006 update. *Nucleic Acids Research*. Jan 1;34(Database issue):D411-4.
- [64] Moreau, Y., Tranchevent, L.C. (2012) Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, 13, 523-526.
- [65] Narayanaswamy, M., Ravikumar, K.E., Vijay-Shanker, K. (2003) A biological named entity recognizer. *Pacific Symposium on Biocomputing*. 427-38.
- [66] NCBI Resource Coordinators. (2014) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*. Jan;42(Database issue):D7-17.
- [67] NCBI Resource Coordinators. (2015) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 43, D6-D17.

- [68] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999) The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. *Stanford InfoLab*.
- [69] Parasuraman, S. (2012) Protein data bank. *Journal of Pharmacology & Pharmacotherapeutics*. 2012 Oct;3(4):351-2.
- [70] Parkinson, H., Kapushesky, M., Shojatalab, M., Abeygunawardena, N., Coulson, R., Farne, A., Holloway, E., Kolesnykov, N., Lilja, P., Lukk, M., Mani, R., Rayner, T., Sharma, A., William, E., Sarkans, U., Brazma, A. (2007) ArrayExpress--a public database of microarray experiments and gene expression profiles. *Nucleic Acids Research*. Jan;35(Databaseissue):D747-50.
- [71] Paro, A. (2013) *Elasticsearch Cookbook*. Packt Publishing.
- [72] Partalas, I., Gaussier, E., Ngomo, A.N. (2013) Result of the First BioASQ Workshop. *In: Proceedings of the first Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop of Conference and Labs of the Evaluation Forum (CLEF 2013)*. Valencia, Span, September 27.
- [73] Pennings, J.L., Koster, M.P., Rodenburg, W., Schielen, P.C., de Vries, A. (2009) Discovery of novel serum biomarkers for prenatal Down syndrome screening by integrative data mining. *PLoS One*. 4(11), e8010.
- [74] Pesquita, C., Faria, D., Falcão, A.O., Lord, P., Couto, F.M. (2009) Semantic similarity in biomedical ontologies. *PLoS Computational Biololgy*. Jul;5(7):e1000443.
- [75] Povey, S., Lovering, R., Bruford, E., Wright, M., Lush, M. and Wain, H. (2001) The HUGO Gene Nomenclature Committee (HGNC). *Human genetics*, 109, 678-680.
- [76] Pruess, M., Kersey P., Apweiler R. (2005) The Integr8 project--a resource for genomic and proteomic data. *In Silicon Biology*. 2005;5(2):179-85.
- [77] Psychogios, N., Hau, D.D., Peng, J., Guo, A.C., Mandal, R., Bouatra, S., Sinelnikov, I., Krishnamurthy, R., Eisner, R., Gautam, B., Young, N., Xia, J., Knox, C., Dong, E., Huang, P., Hollander, Z., Pedersen, T.L., Smith, S.R., Bamforth, F., Greiner, R., McManus, B., Newman, J.W., Goodfriend, T., Wishart, D.S. (2011) The Human Serum Metabolome. *PLoS One*. 6(2), e16957.
- [78] Ran, J., Li, H., Fu, J., Liu, L., Xing, Y., Li, X., Shen, H., Chen, Y., Jiang, X., Li, Y., Li, H. (2013) Construction and analysis of the protein-protein interaction network related to essential hypertension. *BMC Systems Biology*, 12, 7:32.

- [79] Rebholz-Schuhmann, D., Kirsch, H., Arregui, M., Gaudan, S., Riethoven, M. and Stoehr, P. (2007) EBIMed--text crunching to gather facts for proteins from Medline. *Bioinformatics* (Oxford, England), 23, e237-244.
- [80] Rigden, D.J., Fernández-Suárez, X.M., Galperin, M.Y. (2016) The 2016 database issue of Nucleic Acids Research and an updated molecular biology database collection. *Nucleic Acids Research*. Jan 4;44(D1):D1-6.
- [81] Rogers, F.B. (1963) Medical subject headings. *Bulletin of the Medical Library Association*, 51, 114-116.
- [82] Satsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Gaussier, E., Gallinari, P., Artieres, T., Alvers, M.R., Zschunke, M., Gmbh, T., and Ngomo, A.N. (2012) BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. *Information Retrieval and Knowledge Discovery in Biomedical Text*, 2012 AAAI Fall Symposium series.
- [83] Schuler, K.K. (2005) Verbnet: A Broad-coverage, Comprehensive Verb Lexicon. PhD thesis, University of Philadelphia, Philadelphia, PA, USA.
- [84] Takahashi, K., Koike, A., Takagi, T. (2004). Question answering system in biomedical domain. In *Proceedings of the Genome Informatics (GIW 2004)*, 161-162.
- [85] Tan, P.N., Kumar, V., and Srivastava, J. (2000) Indirect association: Mining higher order dependencies in data. In *Principles of Data Mining and Knowledge Discovery*, pages 632-637.
- [86] Tejera, E., Bernardes, J., Rebelo, I. (2012) Preeclampsia: a bioinformatics approach through protein-protein interaction network analysis. *BMC Systems Biology*, 8; 6:97.
- [87] Thompson, P., McNaught, J., Montemagni, S., Calzolari, N., del Gratta, R., Lee, V., Marchi, S., Monachini, M., Pezik, P., Quochi, V., Rupp, C.J., Sasaki, Y., Venturi, G., Rebholz-Schuhmann, D., Ananiadou, S. (2011) The BioLexicon: a large-scale terminological resource for biomedical text mining. *BMC Bioinformatics*. 12:397.
- [88] Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M.R., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., Almirantis, Y., Pavlopoulos, J., Baskiotis, N., Gallinari, P., Artières, T., Ngomo, A.C., Heino, N., Gaussier, E., Barrio-Alvers, L., Schroeder, M., Androutsopoulos, I., Paliouras, G. (2015) An overview

of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*. 16:138.

- [89] Tsatsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Gaussier, E., Gallinari, P., Artieres, T., Alvers, M.R., Zschunke, M., Gmbh, T., and Ngomo, A.N. (2012) BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. Information Retrieval and Knowledge Discovery in Biomedical Text, *AAAI Fall Symposium series*. Jan.
- [90] Tsuruoka, Y., Tsujii, J., Ananiadou, S. (2008) FACTA: a text search engine for finding associated biomedical concepts. *Bioinformatics*. 24(21), 2559-2560.
- [91] Tunstall-Pedoe, W. (2010). True Knowledge: Open-Domain Question Answering Using Structured Knowledge and Inference. *AI Magazine*, 31:80-92.
- [92] UniProt Consortium. (2015) UniProt: a hub for protein information. *Nucleic Acids Research*. Jan;43(Database issue):D204-12.
- [93] Wei, C.H., Harris, B.R., Li, D., Berardini, T.Z., Huala, E., Kao, H.Y., Lu, Z. (2012) Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts. *Database (Oxford)*. Nov 17;2012:bas041.
- [94] Wei, C.H., Kao, H.Y. (2011) Cross-species gene normalization by species inference. *BMC Bioinformatics*. Oct 3;12 Suppl 8:S5.
- [95] Wei, C.H., Kao, H.Y., Lu, Z. (2013) PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*. Jul;41(Web Server issue):W518-22.
- [96] Wishart Research Lab. (2016) FooDB: Food Component Database. <http://foodb.ca/> (access July 11).
- [97] Wishart, D., Arndt, D., Pon, A., Sajed, T., Guo, A.C., Djoumbou, Y., Knox, C., Wilson, M., Liang, Y., Grant, J. et al. (2015) T3DB: the toxic exposome database. *Nucleic Acids Research*, 43, D928-934.
- [98] Wishart, D.S., Jewison, T., Guo, A.C., Wilson, M., Knox, C., Liu, Y., Djoumbou, Y., Mandal, R., Aziat, F., Dong, E. et al. (2013) HMDB 3.0--The Human Metabolome Database in 2013. *Nucleic Acids Research*, 41, D801-807.
- [99] Wolfram Research. (2009) Wolfram Alpha. <http://www.wolframalpha.com/> (accessed July).

- [100] Wren, J.D. (2011) Question answering systems in biology and medicine--the time is now. *Bioinformatics*. Jul 15;27(14):2025-6.
- [101] Yang, H., Swaminathan, R., Sharma, A., Ketkar, V., and D'Silva, J. (2011) Mining biomedical text towards building a quantitative food-disease-gene network. *Studies in Computational Intelligence*. 375:205-225, *Springer*.
- [102] Zhou, D., He, Y. (2008) Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*. Apr;41(2):393-407.
- [103] Zhu, D., Li, D. Carterette, B., Liu, H. (2013) An incremental approach for medline mesh indexing. In: 1st BioASQ Workshop: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. *CEUR Workshop Proceedings*. Aachen, Germany.

Appendices

Appendix A: Description Templates for Drugbank Entries

BioQA uses its Description Generator algorithm to automatically generate short descriptions for biomedical entities retrieved from annotated databases without a description field. The Description Generator first parses a given database entry for information fields and stores the extracted fields in a dictionary. It then generates descriptions by filling in the corresponding blanks in pre-defined sentence templates and producing a description paragraph. Description templates consist of sentence templates grouped into multiple sentence groups. The logic behind Description Generator is discussed in detail in Chapter 5. This section shows sentence templates in groups used to describe a DrugBank [51] drug chemical by first describing its name, synonyms, and drug category (group 1). Then it talks about diseases or conditions that the drug is intended to treat along with the medication's indication (group 2). Then the description talks about brands and manufacturers for the drug, and drug approval information (group 3-5). It further discusses the drug's physical state, melting point and boiling point (group 6), chemical and drug class, mechanism of action (group 7), absorption, half-life, and route of elimination (group 8). Finally, the paragraph describes interactive drugs (group 9), drug targets (group 10), and catalyzing enzymes (group 11).

A.1 Example DrugBank Description Templates

Group 1: Describe a drug's name, list of synonyms, and its primary drug effect.

- DRUG_NAME is also known as DRUG_SYNONYM_LIST.
- DRUG_NAME, also known as DRUG_SYNONYM_LIST, is a DRUG_CATEGORY.
- DRUG_NAME, also known as DRUG_SYNONYM_LIST, is commonly used for its DRUG_EFFECT_LIST effects.
- Known as DRUG_SYNONYM_LIST, DRUG_NAME is most commonly used for its DRUG_EFFECT_LIST effects.

- Commonly used in the treatment of DISEASE_AND_CONDITION, DRUG_NAME is a type of DRUG_CATEGORY drug.

Group 2: Describe which disease and conditions a drug is intended to treat. It's drug category and indications.

- DRUG_NAME is used in the treatment of DISEASE_AND_CONDITION.
- DRUG_NAME is used in the treatment of DISEASE_AND_CONDITION. INDICATION_SYNOPSIS.
- DRUG_NAME is for used in treating DISEASE_AND_CONDITION.
- DRUG_NAME is for used in treating DISEASE_AND_CONDITION. INDICATION_SYNOPSIS.
- DRUG_NAME is approved in treating DISEASE_AND_CONDITION.
- DRUG_NAME is approved in treating DISEASE_AND_CONDITION. INDICATION_SYNOPSIS.
- A type of DRUG_CATEGORY, DRUG_NAME is commonly used in the treatment of DISEASE_AND_CONDITION.
- DRUG_NAME is a type of DRUG_CATEGORY commonly used in the treatment of DISEASE_AND_CONDITION.
- Although most commonly in for the treatment of DISEASE_AND_CONDITION, DRUG_NAME is also sometimes used FOR_INDICATION_SYNOPSIS.
- Although DRUG_NAME is used FOR_INDICATION_SYNOPSIS, it is most commonly indicated for use in the treatment of DISEASE_AND_CONDITION.
- DRUG_NAME is indicated in the treatment of DISEASE_AND_CONDITION, but has also been used FOR_INDICATION_SYNOPSIS.
- DRUG_NAME is indicated for use in the treatment of many conditions, including FOR_INDICATION_SYNOPSIS.
- DRUG_NAME is used FOR_INDICATION_SYNOPSIS.
- Although DRUG_NAME INVESTIGATED_INDICATION_SYNOPSIS, it is most commonly indicated for use in the treatment of DISEASE_AND_CONDITION.
- DRUG_NAME is indicated in the treatment of DISEASE_AND_CONDITION, but has also been INVESTIGATED_INDICATION_SYNOPSIS.

- DRUG_NAME is INVESTIGATED_INDICATION_SYNOPSIS.
- DRUG_NAME is investigated in clinical trials for treating CLINICAL_TRIALS.

Group 3: Describe a drug's brand names.

- DRUG_NAME is branded as BRAND_NAME_LIST.
- Major brands of DRUG_NAME are BRAND_NAME_LIST.
- Known brands of DRUG_NAME are BRAND_NAME_LIST.

Group 4: Describe a drug's various manufacturers.

- DRUG_NAME is manufactured by pharmaceutical companies include MANUFACTUROR_LIST.
- Major manufacturer of DRUG_NAME are MANUFACTUROR_LIST.

Group 5: Describe a drug's approval status with approval country, approval date, and patent ID number.

- DRUG_NAME is approved in APPROVAL_COUNTRY on APPROVAL_DATE (Patent PATTENT_ID).

Group 6: Describe a drug's physical state, melting point, and boiling point.

- DRUG_NAME is a PHYSICAL_STATE.
- DRUG_NAME is a PHYSICAL_STATE with a melting point of MELTING_POINT and boiling point of BOILING_POINT.
- DRUG_NAME is a PHYSICAL_STATE with a melting point of MELTING_POINT.
- In room temperature, DRUG_NAME is a PHYSICAL_STATE with a melting point of MELTING_POINT.
- DRUG_NAME is a PHYSICAL_STATE; its melting point is measured to be MELTING_POINT.
- DRUG_NAME is a PHYSICAL_STATE with a melting point of MELTING_POINT and boiling point of BOILING_POINT.
- DRUG_NAME is a PHYSICAL_STATE with a melting point and a boiling point of MELTING_POINT and BOILING_POINT, respectively.

- DRUG_NAME is a PHYSICAL_STATE with melting points and boiling points of MELTING_POINT and BOILING_POINT, respectively.
- DRUG_NAME is a PHYSICAL_STATE with a melting point of MELTING_POINT.
- DRUG_NAME is a PHYSICAL_STATE with a boiling point of BOILING_POINT.

Group 7: Describe a drug's chemical class, drug class, and mechanism of action.

- DRUG_NAME is a type of CHEMICAL_CLASS_LIST that acts by such mechanism of action: MECHANISM_OF_ACTION_SYNOPSIS.
- DRUG_NAME is a type of CHEMICAL_CLASS_LIST that acts by such mechanism of action: MECHANISM_OF_ACTION_SYNOPSIS. DRUG_CLASS_DESCRIPTION.
- MECHANISM_OF_ACTION_SYNOPSIS. It is a type of CHEMICAL_CLASS_LIST. DRUG_CLASS_DESCRIPTION.
- MECHANISM_OF_ACTION_SYNOPSIS. It is a type of CHEMICAL_CLASS_LIST.
- DRUG_NAME belongs to the chemical class known as CHEMICAL_CLASS_LIST group of drugs, which are known to act via the mechanism of action that MECHANISM_OF_ACTION_SYNOPSIS.
- DRUG_NAME belongs to the CHEMICAL_CLASS_LIST group of drugs, which are known to act via the mechanism of action that MECHANISM_OF_ACTION_SYNOPSIS.
- DRUG_NAME is a type of CHEMICAL_CLASS_LIST, which are known to act via the mechanism of action that MECHANISM_OF_ACTION_SYNOPSIS. DRUG_CLASS_DESCRIPTION.
- DRUG_NAME is a type of CHEMICAL_CLASS_LIST, and is believed to work via the mechanism of action that MECHANISM_OF_ACTION_SYNOPSIS.
- DRUG_NAME belongs to such chemical classes as CHEMICAL_CLASS_LIST. DRUG_CLASS_DESCRIPTION.
- DRUG_NAME belongs to such chemical classes as CHEMICAL_CLASS_LIST. MECHANISM_OF_ACTION_SYNOPSIS.
- DRUG_CLASS_DESCRIPTION.

Group 8: Describes a drug's absorption, half-life, volume of distribution, and route of elimination.

- DRUG_NAME's ABSORPTION and its half-life is HALF_LIFE.
ROUTE_OF_ELIMINATION.
- DRUG_NAME's ABSORPTION and with half life of HALF_LIFE.
- DRUG_NAME has a half-life of HALF_LIFE and its absorption is that ABSORPTION.
- DRUG_NAME has an absorption rate of ABSORPTION along with a half-life of HALF_LIFE.
- DRUG_NAME has an absorption rate of ABSORPTION, a half-life of HALF_LIFE, and a volume of distribution of VOLUME_OF_DISTRIBUTION.
- DRUG_NAME's half-life is HALF_LIFE, while its absorption and volume of distribution are ABSORPTION and VOLUME_OF_DISTRIBUTION, respectively.

Group 9: Describes a drug's interacting drugs.

- It is known that DRUG_NAME interacts with NUM_INTERACTION_DRUGS number of drugs including INTERACTION_DRUG_LIST.
- It is known that DRUG_NAME interacts with INTERACTION_DRUG.
- DRUG_NAME interacts with NUM_INTERACTION_DRUGS number of drugs (INTERACTION_DRUG_LIST).
- DRUG_NAME interacts with NUM_INTERACTION_DRUGS drugs including INTERACTION_DRUG_LIST.
- NUM_INTERACTION_DRUGS drugs are known to interact with DRUG_NAME including INTERACTION_DRUG_LIST.
- It is known that NUM_INTERACTION_DRUGS drugs interact with DRUG_NAME including INTERACTION_DRUG_LIST.
- NUM_INTERACTION_DRUGS drugs interact with DRUG_NAME. These include INTERACTION_DRUG_LIST.
- INTERACTION_DRUG is known to interact with DRUG_NAME.

Group 10: Describes a drug's protein targets.

- DRUG_NAME interacts with target protein TARGET_PROTEIN_LIST.

- DRUG_NAME interacts with target protein TARGET_PROTEIN.
- Known drug targets of DRUG_NAME include DRUG_TARGET_LIST.
- Known drug targets of DRUG_NAME is TARGET_PROTEIN.
- Known drug targets of DRUG_NAME including DRUG_TARGET_LIST.
- DRUG_NAME is known to target DRUG_TARGET_LIST.
- DRUG_NAME is known to target DRUG_TARGET.
- The proteins that DRUG_NAME targets include DRUG_TARGET_LIST.
- DRUG_NAME targets the proteins DRUG_TARGET_LIST.
- DRUG_NAME targets the protein DRUG_TARGET.

Group 11: Describe a drug's catalyzing enzymes.

- Enzymes catalyzing DRUG_NAME includes ENZYME_LIST.
- It is known that DRUG_NAME is metabolized by ENZYME_LIST.
- It is known that DRUG_NAME is metabolized by ENZYME.
- The enzymes known to metabolize DRUG_NAME are ENZYME_LIST.
- The enzyme known to metabolize DRUG_NAME is ENZYME.
- ENZYME_LIST are known to metabolize DRUG_NAME.
- ENZYME is known to metabolize DRUG_NAME.

A.2 Example DrugBank Generated Descriptions

Here are a few examples of DrugBank's original and synthesized descriptions:

DB00680 Moricizine	
Original Description	An antiarrhythmia agent used primarily for ventricular rhythm disturbances. [PubChem]
Generated Description	Moricizine, also known as Moracizinum, Ethmozin, Etmozin, Moracizine, or Moracizina, is an anti-arrhythmia agents and voltage-gated sodium channel blockers. Moricizine is used to treat irregular heartbeats (arrhythmias) and maintain a normal heart rate. Major brands of Moricizine are Ethmazine and Etmozins. This substance is a solid. This compound belongs to the phenothiazines. These are polycyclic aromatic compounds containing a phenothiazine moiety, which is a linear tricyclic system that consists of a two benzene rings joined by a para-thiazine ring. Moricizine has a half-life of 2 hours (range 1.5-3.5 hours). and its absorption is that well absorbed, absorption is complete within 2 to 3 hours Significant first-pass metabolism results in an absolute bioavailability of approximately 38% Administration within 30 minutes after a meal slows the rate, but does not affect the extent of absorption, although peak plasma concentrations are reduced. Moricizine targets the protein sodium channel protein type 5 subunit alpha. An antiarrhythmia agent used primarily for ventricular rhythm disturbances. [PubChem]

DB00686 Pentosan Polysulfate	
Original Description	A sulfated pentosyl polysaccharide with heparin-like properties. [PubChem]
Generated Description	<p>Pentosan Polysulfate is also known as Pentosan sulfuric polyester, Pentosani polysulfas, Pentosanpolysulfat, Pentosano polisulfato, or Pentosane polysulfate. Pentosan Polysulfate is used for the relief of bladder pain or discomfort associated with interstitial cystitis. Major brands of Pentosan Polysulfate are Comfora, Tavan-SP, Elmiron, Hemoclar, Thrombocid, Fibrezym, Fibrase, and Hemoclar. Pentosan Polysulfate is a solid. Pentosan polysulfate is a polymer of xylose hydrogen sulfate and contains two sulfate groups per carbohydrate monomer. Pentosan Polysulfate's slow and with half life of 4.8 hours. Known drug targets of Pentosan Polysulfate include fibroblast growth factor 2, fibroblast growth factor 4, and fibroblast growth factor 1.</p>

DB00689 Cephaloglycin	
Original Description	A cephalosporin antibiotic that is no longer commonly used.
Generated Description	<p>Cephaloglycin, also known as Cefaloglycinum, Cefaloglycine, or Cefaloglicina, is an anti-bacterial agents. Cephaloglycin is used for treatment of severe infections caused by susceptible bacteria. This drug is a solid. This compound belongs to the cephalosporins. These are compounds containing a 1,2-thiazine fused to a 2-azetidinone to for an oxo-5-thia-1-azabicyclo[4.2.0]oct-2-ene-2-carboxylic acid moeity or a derivative thereof. This drug targets the protein penicillin-binding protein 2a.</p>

DB00217 Bethanidine	
Original Description	A guanidinium antihypertensive agent that acts by blocking adrenergic transmission.
Generated Description	Bethanidine, also known as N,N'-dimethyl-N''-(phenylmethyl)-guanidine or Betanidine, is a sympatholytics, antihypertensive agents, and adrenergic agents. Bethanidine is used for the treatment of hypertension. This substance is a solid. This compound belongs to the benzene and substituted derivatives. These are aromatic compounds containing at least one benzene ring. Bethanidine's absorbed rapidly in the gastrointestinal tract following oral administration and with half life of 9 hours (range 7 to 11 hours). Bethanidine is known to target alpha-2b adrenergic receptor, alpha-2a adrenergic receptor, atp-sensitive inward rectifier potassium channel 1, and alpha-2c adrenergic receptor.

DB01326 Cefamandole	
Original Description	Cefamandole (INN, also known as cephamandole) is a broad-spectrum cephalosporin antibiotic. The clinically used form of cefamandole is the formate ester cefamandole nafate, a prodrug which is administered parenterally. Cefamandole is no longer available in the United States.
Generated Description	Cefamandole, also known as Cefadole or Cefamandole nafate, is an anti-bacterial agents. Cefamandole is used for the treatment of serious infections caused by susceptible strains of microorganisms. This medication is a solid. Like all beta-lactam antibiotics, cefamandole binds to specific penicillin-binding proteins (PBPs) located inside the bacterial cell wall, causing the inhibition of the third and last stage of bacterial cell wall synthesis. It targets the protein penicillin-binding protein 2. Cefamandole (INN, also known as cephamandole) is a broad-spectrum cephalosporin antibiotic. The clinically used form of cefamandole is the formate ester cefamandole nafate, a prodrug which is administered parenterally.

DB01325 Quinethazone	
Original Description	Quinethazone (INN, brand name Hydromox) is a thiazide diuretic used to treat hypertension. Common side effects include dizziness, dry mouth, nausea, and low potassium levels.
Generated Description	Quinethazone is also known as Chinetazone, Quinethazon, Quinethazonum, Quinetazona, or Chinethazonum. Quinethazone is used used to treat hypertension. Known brands of Quinethazone are Aquamox, Hydromox, and Idrokin. It is a solid. This compound belongs to the quinazolines. These are compounds containing a quinazoline moiety, which is made up of two fused six-member aromatic rings, a benzene ring and a pyrimidine ring. Known drug targets of Quinethazone including solute carrier family 12 member 1, solute carrier family 12 member 2, carbonic anhydrase 2, solute carrier family 12 member 3, and carbonic anhydrase 1.

DB00318 Codeine	
Original Description	An opioid analgesic related to morphine but with less potent analgesic properties and mild sedative effects. It also acts centrally to suppress cough. [PubChem]
Generated Description	Codeine, also known as Methylmorphine, morphine-3-methyl ether, Morphine monomethyl ether, Codein, or Codeine anhydrous, is a narcotics, antitussive agents, and analgesics, opioid. Codeine is used for treatment and management of pain (systemic). Codeine is a solid. This compound belongs to the morphinans. These are polycyclic compounds with a four-ring skeleton with three condensed six-member rings forming a partially hydrogenated phenanthrene moiety, one of which is aromatic while the two others are alicyclic. Codeine's well absorbed following oral administration with a bioavailability of approximately 90% Maximum plasma concentration occurs 60 minutes post-administration Food does not effect the rate or extent of absorption of codeine and with half life of plasma half-lives of codeine and its metabolites have been reported to be approximately 3 hours. Codeine targets the proteins mu-type opioid receptor, kappa-type opioid receptor, and delta-type opioid receptor. Udp-glucuronosyltransferase 2B4, Cytochrome P450 2D6, Cytochrome P450 3A5, UDP-glucuronosyltransferase 2B7, Cytochrome P450 3A7, and Cytochrome P450 3A4 are known to metabolize Codeine.

DB01452 Heroin	
Original Description	A narcotic analgesic that may be habit-forming. It is a controlled substance (opium derivative) listed in the U.S. Code of Federal Regulations, Title 21 Parts 329.1, 1308.11 (1987). Sale is forbidden in the United States by Federal statute. (Merck Index, 11th ed) Internationally, heroin is controlled under Schedules I and IV of the Single Convention on Narcotic Drugs. It is illegal to manufacture, possess, or sell heroin in the United States and the UK. However, under the name diamorphine, heroin is a legal prescription drug in the United Kingdom.
Generated Description	Heroin is also known as morphine diacetate, diacetylmorphine, or diamorphine. Heroin is used used in the treatment of acute pain, myocardial infarction, acute pulmonary oedema, and chronic pain. This drug is a solid. This compound belongs to the morphinans. These are polycyclic compounds with a four-ring skeleton with three condensed six-member rings forming a partially hydrogenated phenanthrene moiety, one of which is aromatic while the two others are alicyclic. Heroin's bioavailability is less than 35% and with half life of <10 minutes. The proteins that Heroin targets include kappa-type opioid receptor, mu-type opioid receptor, and delta-type opioid receptor. A narcotic analgesic that may be habit-forming. It is a controlled substance (opium derivative) listed in the U.S. Code of Federal Regulations, Title 21 Parts 329.1, 1308.11 (1987). (Merck Index, 11th ed) Internationally, heroin is controlled under Schedules I and IV of the Single Convention on Narcotic Drugs. It is illegal to manufacture, possess, or sell heroin in the United States and the UK.

Appendix B: Automated Paraphrasing Rules

BioQA uses an automated paraphrasing algorithm to transform a synthesized answer to a paraphrased answer. Details for BioQA's paraphrasing algorithm are discussed in Chapter 5. This section shows example paraphrasing rules used by the algorithm to transform sentences. BioQA's automated paraphrasing module achieves paraphrasing results by paraphrasing a paragraph sentence-by-sentence according to a set of predefined, hand-crafted paraphrasing rules. These rules dictate how part of a sentence should be substituted, enumerated, rearranged, or transformed into equivalent expressions. This section shows example paraphrasing rules for B.1) simple substitutions, B.2) substitutions based on word sense, B.3) substitutions based on enumerations, B.4) rearrangement substitutions, B.5) conversion substitutions, and B.6) other substitution rules. Synonym substitution rules with WordNet (English Dictionary words) [62] and PolySearch2's thesauri (Biomedical terms) [52] are not shown here for simplicity.

B.1 Simple Substitution Rules

Phrase substitution Rules

- the town of ↔ the city of
 - the United Kingdom ↔ the UK
 - the United Kingdom ↔ the Great Britain
 - the United States ↔ the U.S.
 - in the vicinity of ↔ in the neighborhood of
 - it is believed that ↔ it is considered that
 - it is endemic to > it is native to
 - it is possible to ↔ it is likely to
 - it is threatened by ↔ it is endangered by
 - on a voyage ↔ on a journey
 - the construction of ↔ the creation of
 - the primary means of ↔ the main way of
 - a collection of ↔ a combination of
 - a couple of ↔ a few
 - a form of ↔ a type of
 - under the command of ↔ under the leadership of
 - under the direction of ↔ under the leadership of
- ... (total 1042 phrase substitution rules)

Simple Substitution Rules

- Every now and then ↔ occasionally
 - Stumble upon ↔ discover
 - Unique ↔ one of a kind
 - Difficult to ↔ hard to
 - Get ↔ obtain
 - In general distribution ↔ widely available
 - Some of these ↔ A few of these
 - Available ↔ obtainable
 - Very small ↔ tiny
 - So good ↔ of such high quality
 - Regularly ↔ routinely
 - Offered ↔ made available
 - Exclusively ↔ selectively
 - Magical ↔ Amazing
 - A masterpiece ↔ superb
 - That is ↔ that's
 - Is truly ↔ is definitely
 - A product ↔ a creation
 - Precision ↔ exacting
 - At its best ↔ without precedent
 - Hand-picked ↔ specially selected
- (Total 853 simple substitution rules)

B.2 Word Sense Substitution Rules

- form (JJ) ↔ type (JJ)
- form (NN) ↔ document (NN)
- transportation (VB) ↔ transport (VB)
- Drag (VB) ↔ haul
- Drag (NN) ↔ burden
- Really (JJ) ↔ very
- Really (RB) ↔ actually
- Result (VB) ↔ arise
- Result (NN) ↔ finding
- Brief (JJ) ↔ concise (JJ)
- roughly (RB) ↔ approximately
- mean (NN) ↔ average (NN)
- Calculating (VB) ↔ determining (VB)
- Calculating (JJ) ↔ conniving (JJ)
- Stirring (VB) ↔ mixing
- Refuses (VB) ↔ declines
- Refuse (NN) ↔ waste (NN)
- Decline (NN) ↔ way down
- Slight (JJ) ↔ small
- Slight (NN) ↔ snub
- causes (NN) ↔ reasons
- causes (VB) ↔ leads to
- cause (NN) ↔ reason
- cause (VB) ↔ lead to
- study (NN) ↔ report
- study (VB) ↔ learn
- state (VB) ↔ say
- crash (VB) ↔ collide
- crash (NN) ↔ collision
- launching (VB) ↔ sending

- Stirring (JJ) ↔ inspiring
- Witness (NN) ↔ observer (NN)
- Witnesses (NN) ↔ observers (NN)
- Witness (VB) ↔ observe
- Refuse (VB) ↔ decline
- Experience (NN) ↔ overview
- Addition (VB) ↔ adding
- launching (NN) ↔ initiation
- Expressive (JJ) ↔ evident
- Blend (NN) ↔ mixture
- Pretty (RB) ↔ very
- Beverage ↔ drink (NN)
- Addition (NN) ↔ arrival
- Leader (JJ) ↔ ahead (JJ)
- Leader (NN) ↔ boss
- That (NN) ↔ this
- The cause (NN) ↔ The reason for this
- Cause (VB) ↔ generate
- absent (VB) ↔ without
- murder (VB) ↔ kill
- throughout (JJ) ↔ around
- Blend (VB) ↔ mix
- Try (NN) ↔ attempt
- Meeting (NN) ↔ conference
- Support (VB) ↔ hold up
- Use (NN) ↔ application
- Function (VB) ↔ play a role
- First (NN) ↔ number one
- Show off (NN) ↔ attention grabber
- That (RB) ↔ which
- Cause of (NN) ↔ reason for
- Cause (NN) ↔ source
- absent (NN) ↔ gone
- murder (NN) ↔ homicide
- throughout (NN) ↔ everywhere
- Beloved (VB) ↔ much loved
- Pretty (JJ) > beautiful
- Try (VB) ↔ make an effort
- Meeting (VB) ↔ connecting
- Support (JJ) ↔ backing
- Use (VB) ↔ employ
- Function (NN) ↔ role
- First (JJ) ↔ initial
- Show off (VB) ↔ Draw attention to him/herself
- Accepting (VB) ↔ taking in
- Relative (JJ) ↔ comparative
- Form (VB) ↔ make
- Fail (VB) ↔ not succeed
- Fall (NN) ↔ autumn
- Fall (NN) ↔ drop
- Accepting (NN) ↔ tolerant
- Relative (NN) ↔ family relation
- Form (NN) ↔ shape
- Fall (VB) ↔ drop
- Pioneer (VB) ↔ lead the way
- Pioneer (VB) ↔ open up

- Pioneer (NN) ↔ forerunner
- Testing (JJ) > taxing
- Experienced (NN) ↔ veteran
- Prevent (VB) ↔ stop (VB)
- Testing (VB) ↔ assessing
- Experienced (VB) ↔ witnessed
- Present (VB) ↔ show

B.3 Enumeration Rules

- noun1, noun2 and noun3 ↔ noun3, noun1 and noun 2
- Adjective 1 and adjective 2 ↔ adjective 2 and adjective 1
- Adjective1, adjective2 and adjective3 ↔ adjective3, adjective1 and adjective2

B.4 Rearrangement Rules

- some XXXs ↔ several XXXs
- cost of XXX ↔ XXX prices
- Looking forward to XXXing ↔ Hoping to XXX
- Over XX ↔ more than XX (where XX is a number)
- The XXXion of ↔ XXXing
- XXX (noun) department ↔ department of XXX
- XXX (noun) department ↔ department for XXX
- XXX (noun) faculty ↔ faculty of XXX
- XXX (noun) office ↔ office of XXX
- XXX (noun) office ↔ office for XXX
- The [Adj] of the [Noun] ↔ the [Noun]'s [Adj]
- XXX said "QUOTEBODY" ↔ "QUOTEBODY" said XXX
- XXX said "QUOTEBODY" ↔ XXX noted "QUOTEBODY"

B.5 Conversion Rules

- Convert XXX feet to YY meters
- Convert XXX pounds to YY kilograms
- Convert XXX ft to YY meters

- Convert XXX lbs to YY kilograms
- Convert XXX inches to YY centimeters
- Convert XXX in. to YY cm.

B.6 Other Rules

- Never change anything in quotes
- Never change proper nouns, acronyms or names (terms with upper case letters in the first letter position)

Appendix C: Other Information Extraction Techniques in BioQA

In this section, I discuss BioQA's approach for recognizing chemical terms from free text documents, and another approach for extracting attributes for biomedical terms based on PolySearch2 [52]. Both algorithms are used in the construction of BioKB, the knowledge base component for BioQA. C.1 discusses a chemical term recognition algorithm used to parse chemical names from surface text (expressions that are actually used in a sentence). C.2 discuss an approach to automatically extract attributes from text.

C.1 Chemical Term Recognition

Recognizing chemical names from text is challenging as thesauri for chemical names will never be complete as new compounds are constantly being discovered and synthesized. In developing BioKB, we developed a chemical term recognizer which is capable of recognizing IUPAC and IUPAC-like chemical names from text. This chemical term recognizer uses a hybrid approach. Given a text paragraph, it first identifies chemical names using strict dictionary match with a unified name thesaurus generated by combining Jochem [39], PubChem, DrugBank [51], and HMDB [98] names/synonyms. To extract terms that are not present in our chemical thesaurus, the chemical name recognizer generates candidate terms by removing all words appearing in a general English dictionary, as well as punctuation marks. Candidate terms are then classified by a binary Support Vector Machine classifier, trained using N-character substring features of chemical names and synonyms in our chemical thesaurus in contrast with words in a general English dictionary. Each term is classified as being IUPAC-like or not, and then the compound term is assigned a score based on the number of IUPAC terms it contains. Finally, an empirical cut-off is used to select compound terms that are most likely to be chemical names. Using this chemical name recognizer, we were able to identify 120+ novel compounds that are mentioned to be present in urine but not yet captured in the previous version of HMDB [98]. Additionally, we were also able to confirm 500+ urine compounds that are already in the current version of HMDB. The chemical name recognizer is still imperfect as it picks up species

names and medical procedures as these terms have not been included as negative examples in the original training set.

C.2 Attribute Extraction

Attribute extraction is yet another important task central to biomedical information extraction. For example: 1) given a name for a species or genus, we would like to extract its phenotype from a collection of reference databases, 2) given a name for a compound, we would like to extract its health effects from a collection of free-text. In developing BioKB, we directed a great deal of effort in extracting attributes for biomedical entities from text. In this approach, we customized PolySearch2 [52] to each of the information extraction tasks. We first search the literature (MEDLINE, PubMed Central articles, etc.) using a target term's name and synonym as search keyword, and then scan relevant text snippets to target the term of interest (a predefined list of potential attributes). Finally, an empirical cut-off is applied to the final result so only strong associations are considered for further refinement. In this section, we showcase the approach for attribute extraction based on the PolySearch2 [52] association finding algorithm. Furthermore, we illustrate the approach in action for extracting phenotypic information for prokaryotes and health effects for food metabolites.

Prokaryotes are a kingdom of microbes that include both eubacteria and archaeobacteria. Phenotypic information for bacteria and archaea are scattered in various bioinformatics databases with different formats and different levels of coverage. In a recent effort to consolidate phenotypic information for all known prokaryotes (bacteria and archaea), we mined more than a dozen online databases and compiled the most comprehensive bacterial phenotype database to date. Furthermore, missing data in the phenotype database was calculated from information contained in sequenced bacteria genomes, inferred from biochemical pathways, and extrapolated from closely related species along branches of the phylogenetic tree. Using the above methods, we successfully increase the percent coverage of all data fields from 55.30% to 65.11% with text mining and calculations, and finally to 86.92% with taxonomic extrapolation. Table 20 details the percent coverage of 14 data fields after initial data wrangling (initial coverage), text mining, and extrapolation using taxonomic relations in the NCBI taxonomy. The resulting phenotypic

database contains comprehensive phenotypic information for 10,835 prokaryote species and strains. This phenotype database contains 38 data fields, including oxygen requirements, gram stain, cell shape, motility, temperature range, metabolism, energy sources, associated diseases and pathogenicity, just to name a few. Information in this database are integrated in BacMap [21], an up-to-date electronic atlas of annotated bacterial genomes, and METAGENassist [4], an analytical pipeline for comparative meta-genomic studies.

Data Field	Initial Coverage (%)	Text Mining Coverage (%)	Taxonomic Extrapolation Coverage (%)
Gram Stain	69.00	71.09	94.68
Cell Shape	83.56	87.02	97.99
Motility	50.50	53.26	90.95
Human Pathogen (Y/N)	17.58	18.76	68.10
Oxygen Requirement	61.51	63.51	93.38
Temperature Range	37.27	39.20	82.06
Symbiotic (Y/N)	20.02	21.35	71.23
Habitat	78.49	81.53	96.63
Host Name	11.19	12.06	52.57
Cell Arrangement	36.80	38.96	78.35
Sporulation (Y/N)	16.63	17.76	61.94
Energy Source	33.81	36.30	81.88
Metabolism	3.42	57.26	91.22
Disease Association	2.03	96.38	99.34
Total	55.30	65.11	86.92

Table 20: Percent (%) coverage for selected data fields in the prokaryotic phenotype database in BacMap and MetaGenAssist. The phenotype database contains a total of 38 data fields (14 shown here) for 10,835 prokaryote species, subspecies and strains.

Metabolite	Health Effects	Score	Num. Ref	Evidence
Curcumin	anti-oxidant	2454	278	(PMID: 20508869) ... indicating that the potent antioxidant curcumin can be used as an adjuvant in antiepileptic therapy.
Curcumin	anti-inflammatory	1291	155	(PMID: 17569207) in this review, we describe both antioxidant and anti-inflammatory properties of curcumin, ...
Curcumin	anti-cancer	759	92	(PMID: 20655375) the present study indicated the potential of tf-c-sln in enhancing the anticancer effect of curcumin in breast cancer cells in vitro.
Curcumin	anti-tumor	466	56	(PMID: 16364242) the induction of growth-arrest and apoptosis ... suggests this be a mechanism by which curcumin induces antitumor activity in t cell leukemia.
Curcumin	apoptotic	417	50	(PMID: 20138829) in this study we found that curcumin induces apoptotic cell death in mcf-7 cells ...
Curcumin	Neuroprotectant	389	37	(PMID: 16075466) these findings attribute the neuroprotective effect of curcumin against i/r-induced neuronal damage...
Curcumin	anti-depressant	108	7	(PMID: 19882093) curcumin can be a useful antidepressant especially in cases which respond to drugs having mixed effects
Curcumin	anti-viral	96	12	(PMID: 21299124) thus, our results suggest an important antiviral effect of curcumin wherein it potently inhibits coxsackievirus replication ...
Curcumin	anti-fungal	59	6	(PMID: 17199240) these results indicate an antinociceptive activity of resveratrol and curcumin ...

Table 21: An example of potential health effects extracted from MEDLINE abstracts for curcumin, a phytochemical found in the popular Indian spice turmeric. This table lists examples of potential health effect (extracted using the in-house attribute extractor), their scores in co-occurrence analysis, and supporting evidence from reference publications.

We also applied similar techniques to extract health effects for food metabolites. Over the past few years we have developed a health effect annotator to mine health effects, food tastes, and food functions from MEDLINE abstracts for 42,000+ food metabolites that are being annotated in the FooDB project. More specifically, the health effect annotator takes a list of compound names/synonyms and a manually curated health effect thesaurus as input, and then searches MEDLINE abstracts for co-occurrences of health effect terms and compound names using a customized PolySearch algorithm [16, 17]. Similar to PolySearch2 [52], the association between compounds and health effects are scored and ranked by the frequency of term co-occurrence. Tighter co-occurrences are given higher scores. We found that specific attention should be given to the conclusion part of MEDLINE abstracts, as co-occurrences of compound names and health effect terms often signifies a conclusive statement of the association. Table 21 shows a few examples of the extracted potential health effects of *curcumin*, a phytochemical found in the popular Indian spice turmeric. Same analysis has been conducted on more than 24,000 compounds in FooDB [96].

We are working to expand the health effect annotator to be a general attribute learner that takes an arbitrary biomedical term, a set of thesaurus terms, and extracts descriptive attributes of an entity. Extracted attributes can be used to generate descriptions for a biomedical entity based on a certain template. For example, a statement regarding the health effects of curcumin can be synthesized as “Curcumin has been shown to exhibit anti-oxidant, anti-inflammatory, anti-cancer, anti-tumor, apoptotic, neuroprotectant, anti-depressant, anti-viral, anti-fungal, and immunomodulator effects.” Many descriptions in ECMDB [35], HMDB [98], and FooDB [96] are generated or enriched using this method.