Question Answering for Alzheimer Disease using Information Retrieval

Sanmitra Bhattacharya[1,2] and Luca Toldo[2]

[1]Department of Computer Science, The University of Iowa, Iowa City, IA, USA
[2]Knowledge Management, Merck KGaA, Darmstadt, Germany
sanmitra-bhattacharya@uiowa.edu
luca.toldo@merckgroup.com

Abstract. With the tremendous growth of biomedical literature and data, it's no longer feasible for researchers to manually sift through this information for answering questions on specific topics. The "Machine Reading of Biomedical Texts about Alzheimer Disease" task of CLEF QA4MRE encouraged the development of systems that can automatically find answers to questions on Alzheimer disease. To this end, we developed several information retrieval(IR) and semantic web-based strategies. Our best performing strategy used a combination of query processing followed by IR on the background corpus, distributed by the organizers, to find correct answers. Using our systems, the highest cumulative and individual c@1 scores achieved were 0.47 and 0.66 respectively.

Keywords: Question answering, information retrieval, semantic web

1 Introduction

Research in the biomedical domain has seen unprecedented growth in the past few decades. While the majority of biomedical research is still conducted in wet-labs on plant and animal models i.e. *in vivo*, there has been a steady rise in the use of automation and *in-vitro* techniques in this area. The result is an ever-increasing volume of data and literature which can no longer be processed manually. Making sense of this insurmountable amount of data and information without computational techniques is practically impossible for life sciences researchers. To this end, various biomedical text mining applications have been developed to fulfill the information need for life sciences researchers. Information retrieval and automated question answering (QA) are essential examples of such applications which have found increasing importance among researchers.

Biomedical information retrieval is a thriving domain. Retrieval systems like PubMed cater to the information need of thousands of users every day. In 2011 more than 1.8 billion PubMed searches were conducted, an increase of 13% from the year before[1]. While such retrieval systems facilitate the search for documents relevant to users' queries, it does not provide precise answers to questions. In

[1] http://www.nlm.nih.gov/bsd/bsd_key.html

contrast, QA takes a fine grained approach to information retrieval in providing precise answers to questions. In essence, it can be viewed as an information retrieval problem where the task is to find sections (phrases/sentences/paragraphs) of an article that are relevant to a question rather than finding the entire article[32].

QA in the biomedical domain has two major challenges. First, entities involved in the question can have synonyms, abbreviations and various sources of ambiguity which makes the search process challenging. Second, in practical settings, with the availability of millions of biomedical articles, medical records and domain-specific thesauri, answers to questions can vary widely depending on the domain under discussion [17]. Thus, it is well recognized that QA in the medical/clinical domain is quite distinct from that of biological domain [25].

The "Machine Reading of Biomedical Texts about Alzheimer Disease" task of CLEF QA4MRE aimed at exploring and evaluating systems designed for answering questions about Alzheimer disease. Similar to QA tasks in the biological domain, the goal was to find precise answers to questions from one or more corpora of biomedical texts on Alzheimer disease. In our implementation of systems to address this problem, we followed various information retrieval-based sentence extraction approaches in finding the most suitable answer to a given question. The rest of the paper is organized as follows: Section 2 reviews some of the related literature in the domains of medical as well as biological QA, Section 3 discusses the details of the task and the datasets, Sections 4 and 5 elaborate on the various strategies used for our submitted and un-submitted runs along with the results, and finally, in Section 6 we outline future work in this direction.

2 Related Research

Automated QA has undergone tremendous progress in the recent years. QA systems such as IBM's 'Watson' gained prominence in popular culture through participation and eventually winning the 'Jeopardy! Challenge'[2]. 'Watson' used Apache UIMA's real-time content analytics[3] in conjugation with deep natural language processing, information retrieval, machine learning, etc. to provide answers to open domain questions in an extremely efficient way. Application of such advanced systems to QA in specialized domains such as Alzheimer disease would be quite interesting. Other than such domain independent QA systems there has been considerable research on QA for medical and biological domains.

Evidence-based medicine (EBM) is one of the primary motivators for medical/clinical QA [35]. EBM follows a well-studied PICO (Problem/ Population, Intervention, Comparison and Outcome) framework for structuring questions [20]. One of the drawbacks of PICO is that it is well-suited for answering intervention-specific questions but less suitable for answering other clinical information needs [12].

[2] http://www.jeopardy.com/
[3] https://blogs.apache.org/foundation/entry/apache_innovation_bolsters_ibm_s

Several systems have been developed over the years for clinical QA. Most systems use some domain specific knowledge for finding answers to questions. Following the PICO standards, [10] proposed a system that combines knowledge extraction from MEDLINE abstracts with document re-ranking for improving performance of their EBM-based QA system. Several machine learning based systems have been proposed for identifying questions along various dimensions such as *answerable* or *unanswerable* questions, definitional questions ("What is X?") or categorical questions (e.g. etiology, procedure, and diagnosis), respectively [34, 33, 6]. Some systems use pattern based semantic models and UMLS concepts, semantic types and relationships for extracting answers from MEDLINE abstracts [7]. Several studies have proposed hybrid approaches based on information retrieval and summarization (using UMLS semantic types) for extracting candidate answers to given questions [8, 9]. More recently, [5] proposed an online clinical QA system called AskHERMES using machine learning techniques on textual, syntactic and UMLS-based semantic features, derived from questions, to form extractive summaries from candidate documents as answers.

The strategies for QA in the biological domain mimics the practices of clinical QA (except for the PICO framework). Semantic-based approaches use the UMLS metathesaurus and other thesauri for query expansion strategies and shortlisting candidate articles as answers [28]. Similar to the clinical QA system of [8], [24] proposed a biomedical QA system based on automated summarization of documents relevant to a particular question. Few studies have also proposed application-specific QA systems. For example, [15] proposed a QA system for bio-molecular events. Their approach uses semantic role labeling and semantic graph-based sentence extraction followed by several post-processing steps to generate summaries that answer specific questions.

Similar to recent research on finding important segments of biomedical text for document summarization [3], here we explore various information retrieval-based strategies to identify and rank the most relevant sentences to a given question and thereby identify the correct answer to a particular question.

3 Description of the Task

3.1 Objective

The objective of the "Machine Reading of Biomedical Texts about Alzheimer Disease" task, as the title suggests, is to explore various strategies of machine reading systems to answer questions pertaining to Alzheimer disease[4]. In comparison to other domains, biomedical text has its unique challenges (as discussed before). Over the past couple of decades various tools and applications have been designed for so-called micro-tasks on biomedical text, namely, information extraction [14], named-entity recognition (NER) [11], relationship extraction [4], event extraction [13], etc. Various marco-tasks have also been built on top of these micro-tasks such as retrieval of documents for particular genes

[4] http://celct.fbk.eu/QA4MRE/index.php?page=Pages/biomedicalTask.html

[1], literature-based discovery [19], entity-based summarization [16], question answering [15], etc.

In the Alzheimer QA task the focus is on reading single documents and identification of answers for a set of questions using information implicitly or explicitly stated in the text. Systems developed for this task are required to identify a correct answer from a set of 5 probable answers for a question in a Multiple Choice Question (MCQ) setting. To identify the correct answers systems may use a reference document collection on Alzheimer disease provided by the Lab organizers. It is important to note here that answers to questions pertaining to a particular document can be detected using that document only, while systems may benefit from using the background collection and associated pre-processed information made available to the participants.

3.2 Datasets

PubMed abstracts A set of 66,222 abstracts relevant to Alzheimer disease was obtained using PubMed search and made available to the participants.

Full text articles from PubMed Central (PMC) A set of 8,249 Open Access full text articles were obtained from PMC in PDF format. 7,512 of these articles were converted into text format using LA-PDFText[5].

A smaller set of 1,041 full text articles in HTML and text format from the last three years on Alzheimer disease was also obtained from PMC.

Elsevier full text articles A set of 379 full text articles and 103 abstracts were obtained from Elsevier in XML and text format. This set, containing articles referring to 45 core hypothesis in Alzheimer disease, was manually selected by an expert in this area.

Annotated Data The documents of the background collection were annotated across different dimensions using various publicly available tools. The dependency parser, GDep [22] was used for annotating words, lemmas, chunks, parts-of-speech (POS), named-entities (NE), parent nodes in the dependency syntax trees and dependency syntax labels. The popular biomedical NE tagger ABNER [23] along with another UMLS-based NE tagger developed at CLiPS were also used for NE annotation.

Training Data The training set comprised of a single full-text XML-formatted article along with questions and answers in the MCQ format described above. Correct answers to all the questions were also made available to the participants.

[5] http://code.google.com/p/lapdftext

Test Data The test set comprised of 4 full-text documents, each containing 10 MCQ questions in a format similar to the training set. Both training and test documents were processed with the annotation strategy outlined above.

Questions in the training and test set could be classified into 3 degrees of difficulty, namely, *simple* (answer present almost verbatim in the article), *medium* (questions containing lexico-semantic alienations of NEs), *complex* (reasoning and derivation-based questions). The type of questions spanned across various topic types such as identification of experimental evidences, protein-protein interactions, gene synonymy relations or regulatory relations, in the context of Alzheimer disease.

4 Methods

4.1 Question-Answer Pre-processing

Greek alphabet expansion In the first step we expand all Greek alphabets into their corresponding English names. For example, 'Aβ' is converted into 'Abeta'.

Dictionary generation In this step we parsed the Elsevier articles to identify expanded forms of abbreviations (marked by 'QUALIFIER Abbreviations' in the document collection). A dictionary of abbreviations and their corresponding full forms was created and applied to the questions and answers of the test ocuments.

POS tagging In this step we used the Stanford Log-linear POS tagger [31] to identify all terms tagged as nouns, adjectives, adverbs, symbols, cardinal numbers, and select verbs forms (base forms and past participles).

4.2 Document Processing

Sentence splitting We used the GENIA sentence splitter [21] for all sentence splitting mechanisms in our systems. We split the documents of the test set and background collection (except the larger PMC full-text collection) into sentences using this tool. However, the output from the sentence splitter had to be fixed because of incorrect sentence splitting based on certain words that appear frequently in the scientific literature (such as 'Fig.').

Indexing We created several indexes from the test document and the background collection. The Indri information retrieval tool [26] was used for indexing the documents. The Krovetz stemmer was used for all indexing experiments.

Test document index: Each test document was indexed individually at the sentence level for future retrieval purposes. This index is referred to as *TestIndex* in the remainder of this paper.

Background collection index:

- Elsevier index: All sentences from the Elsevier corpus were indexed using the Indri IR tool. A total of 101,778 sentences from 482 documents were indexed.
- PMC full-text index (smaller): We indexed all sentences from the smaller PMC full-text corpus using Indri. A total of 854,034 sentences from 1,041 documents were indexed.
- PubMed abstract index: All sentences from the PubMed abstract corpus were indexed using Indri. A total of 599,060 sentences from 66,222 documents were indexed.

A single index was created by combining the above 3 indexes. This is referred to as *CombinedIndex* in the remainder of the paper.

4.3 Submitted Runs

We submitted 7 official runs for this task. In this section we highlight the major strategies underlying each run. For all runs except two we followed a two step retrieval approach. Generally, in the first step we retrieved a set of candidate sentences that may contain the correct answer to a question. In the second step a single correct answer from this pool of candidates is selected using retrieval techniques. For two of the submitted runs we used a slightly different strategy. In one strategy we used a hypothesis generation technique using both the questions and answers of QA task for finding the correct answers. In the other strategy we followed a majority voting scheme for selecting the correct answer from a pool of runs.

Run 1 In this run we used the preprocessed questions (i.e. questions processed using the steps outlined in Section 4.1) for retrieving a candidate set of sentences that might contain the correct answer. The retrieval of sentences was done only on the test document for that question using Indri's default language model. For querying we used Indris belief operator (#combine) which, unlike Boolean operators (e.g. AND, OR, etc.) that returns only binary values, weighs each term equally and prioritizes documents (sentences in this case) containing more query terms. Stop words were removed for every retrieval experiment. We limited our retrieval to the top 5 sentences from the *TestIndex*.

In the second step we followed the same pre-processing steps as before but only on the answers. Again we used Indri's #combine operator to retrieve the correct answer for each question. An answer choice that's also a part of a question was automatically discarded as a candidate. To identify the correct answer to a question we count the number of sentences (out of 5) that are retrieved for each answer. The answer that retrieved the most number of sentences was considered as the correct one. In case of a tie, we did not answer that particular question.

Run 2 In this run we followed the steps identical to Run 1 except that we used a tf*idf retreival model instead of Indri's default language model.

Run 3 This run is also similar to Run 1 but here we did not limit our retrieval to the top 5 candidate sentences in the first step. As a result, the retrieval in the second step is based on a larger pool of sentences from which we select the correct answer as the one retrieving the most number of sentences from the candidate pool. Similar to Run 2 we use the tf*idf retrieval model instead of the default language model of Indri. In case of a tie, we skip answering that question.

Run 4 In this run, we adopted the same strategy as in Run 3 in the first step. However in the second step, we selected a correct answer by ranking the retrieved sentences by tf*idf retrieval scores and selecting the answer corresponding to the highest scoring sentence as the correct one.

Run 5 In this run we followed similar strategies as in Run 2, but instead of selecting the correct answer by the count of retrieved documents we used the highest retrieval score for identifying the correct answer.

Run 6 This run followed a considerably different strategy compared to all of the previous runs. Here each preprocessed question was combined with all possible answers to that question to form various hypotheses. For example the question "Which technique was used to determine the cellular locations of the CLU1 and CLU2 gene products?" has 5 probable answers, such as "intracellular and secreted", "ER", etc. In this strategy we combined the question and the answers into a single hypothesis. For each question we created 5 hypotheses which were tested for validity using retrieval strategies adopted in our system.

In contrast to the previous runs, here we used only the background collection index, *CombinedIndex*, for retrieval of sentence. We limited our retrieval to only the highest scoring sentences. One of the five hypotheses which fetched the highest scoring sentence was identified as the correct answer. In case of a tie we did not answer that question.

Run 7 In order to combine the retrieval results from Runs 1-5 we employed a majority-based voting to identify the correct answers. Answers that were voted most frequently as correct ones in the Runs 1-5 were selected as the correct answers for this run. In case of a tie we did not answer that question.

4.4 Unsubmitted Runs

Other than the submitted runs, we tried various other strategies on the training document with slightly poor performances. These runs were not included in the submission system. For all unsubmitted runs we generated 50 pseudo-sentences following the hypothesis generation strategy of Run 6, combining the query with each possible answer. The training document was (automatically)

Fig. 1. The pie charts show for each run the number of questions that Answered Right, Answered Wrong and Unanswered. We did not have any Unanswered Right/Wrong instances.

split into individual sentences, and then both training sentences and pseudo-sentences were tagged with UIMA-based Luxid® 6[6] Biological Entity Relation (BER) Skill Cartridge and the Medical Entity Relation (MER) Skill Cartridge, two rule and dictionary-based high precision taggers. We then selected the metric that maximized the number of right answers. The steps outlined in the following sub-sections are based roughly on the semantic search strategy used for electronic medical record retrieval [30] for the 2012 TREC Medical Records track (TRECMED).

Luxid® 6 shallow linguistic similarity In this run we used the highest linguistic similarity score computed by Luxid® 6, using as features the shallow linguistic entities computed using the Luxid® 6 Analytics2 Skill Cartridge, to find the correct answer. It returned 30% correct answers on the training set.

Luxid® 6 semantic similarity In this run we used the highest semantic similarity score computed by Luxid® 6, using as features the tags provided by

[6] http://www.temis.com/

the BER and MER taggers, as metrices to find the correct answer. It returned 40% correct answers on the training set.

Luxid® 6 Crossmatch method In this run we used the highest number of edges between the the test document and the pseudo-sentences (computed using Luxid's Crossmatch), as a way to find the most suitable answer. This method is an approximation of the Literature Based Discovery method of Swanson [27]. It delivered 40% correct answers.

Naïve Bayesian classifier following network analysis For this run we trained a Naïve Bayesian classifier using as features the network descriptors [29] computed using KNIME [2]. This method had no predictive power in the training set.

5 Results

Figure 1 shows the proportion of questions (total 40) that were answered along with their accuracies for each run. For example, in Run 1 our system answered 32 questions while the remaining 8 were unanswered. Out of the answered questions, 12 were correct and the remaining 20 were incorrect. Similarly, for Run 3 we answered all the questions but only a third of them were correct and the remaining were incorrect. The best performing run was Run 6 where we answered 36 questions while the remaining were unanswered. Out of the answered questions 17 were judged correct while the rest (19) were incorrect.

For a quantitative evaluation that takes into account the categories of answered correct, answered incorrect and unanswered questions, the c@1 metric [18] was used. The metric rewards systems that reduces the number of incorrect answers while maintaining the number of correct answers by not answering some questions. It is represented using Equation (1), where nr is the number of correctly answered questions, nu is the number of unanswered questions, and n is the total number of questions.

$$c@1 = \frac{(nr + nu \times (\frac{nr}{n}))}{n} \tag{1}$$

Table 1 shows the overall c@1 scores for all the submitted runs. Run 6, based on hypothesis generation from questions and answers and using only the background corpus for retrieval of correct answers, performs the best with a c@1 score of 0.47. On the other hand, Run 3 performs worst with a c@1 score of 0.25. The scores for this run are largely affected due to the exhaustive answering of all given questions.

In addition to the overall c@1 measure we also calculate the c@1 scores for the individual reading tests, as shown in Table 2. In this table, we find that while Run 1 performs the best for reading test 2 (0.66), other runs such as Runs 4, 6 and 5 perform well for reading tests 1 (0.30), 4 (0.60) and 3 (0.48),

Table 1. Overall c@1 measures for submitted runs

Runs	c@1 scores
Run 1	0.36
Run 2	0.40
Run 3	0.25
Run 4	0.26
Run 5	0.39
Run 6	**0.47**
Run 7	0.35

respectively. However, the best performing system, Run 6, surpasses other runs in terms of mean and median scores with a moderate standard deviation. It is also interesting to note that on average, submitted runs for reading test 2 perform considerably better than others.

Table 2. c@1 scores for individual reading tests (r_id 1-4)

Runs	r_id 1	r_id 2	r_id 3	r_id 4	Median	Mean	S.D.
Run 1	0.00	**0.66**	0.22	0.44	0.33	0.33	0.28
Run 2	0.24	0.60	0.30	0.44	0.37	0.40	0.16
Run 3	0.00	0.40	0.30	0.30	0.30	0.25	0.17
Run 4	**0.30**	0.11	0.40	0.22	0.26	0.26	0.12
Run 5	0.00	0.50	0.44	**0.60**	0.47	0.39	0.27
Run 6	0.22	0.60	**0.48**	0.55	0.52	0.46	0.17
Run 7	0.00	0.55	0.36	0.44	0.40	0.34	0.24

6 Conclusion

In this paper we proposed various strategies for automated question answering for Alzheimer disease. We selected an information retrieval based approach for this purpose. In the seven submitted runs we tried two basic approaches with some finer modifications in each run. In the first approach we tried a two-step retrieval process where in the first step we select a pool of candidate sentences and in the next step we select the correct answer from this pool. This is done either by the retrieved sentence count or by the highest retrieval score. In the second approach we implemented a hypothesis generation technique using both the questions and answers of the reading tests, followed by a retrieval score based answer selection process. The most notable difference between these two approaches is that the first one uses only the test corpus for selecting the correct answer while the second approach uses only the background corpus. Incidentally we find that the second approach performs considerably better than the first

one. Also a combination strategy for the various runs based on the first approach performs worse than the second approach.

Further, we find that there is significant variability in the performance of the various systems. Four different systems provide the top scores for the four reading tests. This shows the potential for leveraging the best results from the different strategies using a unifying technique, more sophisticated than the simple majority voting strategy used in one of our submitted runs.

In future work we would like to explore other semantic web-based techniques (similar to the ones outlined in Section 4.4) in aiding the performance of our retrieval-based systems. We would also like to explore other techniques for correct answer selection from the candidate answer pool. Finally, in our current implementation use of pre-processed information provided by the organizers was limited and largely out of scope. We would like to implement strategies that benefit from these information in future work.

References

1. C. N. Arighi, P. M. Roberts, S. Agarwal, S. Bhattacharya, G. Cesareni, A. Chatr-Aryamontri, S. Clematide, P. Gaudet, M. G. Giglio, I. Harrow, E. Huala, M. Krallinger, U. Leser, D. Li, F. Liu, Z. Lu, L. J. Maltais, N. Okazaki, L. Perfetto, F. Rinaldi, R. S?tre, D. Salgado, P. Srinivasan, P. E. Thomas, L. Toldo, L. Hirschman, and C. H. Wu. BioCreative III interactive task: an overview. *BMC Bioinformatics*, 12 Suppl 8:S4, 2011.
2. M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007.
3. S. Bhattacharya, V. Ha-Thuc, and P. Srinivasan. MeSH: a window into full text for document summarization. *Bioinformatics*, 27(13):i120–128, Jul 2011.
4. C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: protein-protein interactions. *Proc Int Conf Intell Syst Mol Biol*, pages 60–67, 1999.
5. Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J. J. Cimino, J. Ely, and H. Yu. AskHERMES: An online question answering system for complex clinical questions. *J Biomed Inform*, 44(2):277–288, Apr 2011.
6. Y. G. Cao, J. J. Cimino, J. Ely, and H. Yu. Automatically extracting information needs from complex clinical questions. *J Biomed Inform*, 43(6):962–971, Dec 2010.
7. T. Delbecque, P. Jacquemart, and P. Zweigenbaum. Indexing umls semantic types for medical question-answering. *Studies In Health Technology And Informatics*, 116:805–810, 2005.
8. D. Demner-Fushman, B. Few, S. E. Hauser, and G. Thoma. Automatically identifying health outcome information in MEDLINE records. *J Am Med Inform Assoc*, 13(1):52–60, 2006.
9. D. Demner-Fushman and J. Lin. Answer extraction, semantic clustering, and extractive summarization for clinical question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 841–848, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

10. D. Demner-Fushman and J. Lin. Answering clinical questions with knowledge-based and statistical techniques. *Comput. Linguist.*, 33(1):63–103, Mar. 2007.
11. L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6 Suppl 1:S1, 2005.
12. X. Huang, J. Lin, and D. Demner-Fushman. Evaluation of PICO as a knowledge representation for clinical questions. *AMIA Annu Symp Proc*, pages 359–363, 2006.
13. J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 1–9, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
14. M. Krallinger and A. Valencia. Text-mining and information-retrieval services for molecular biology. *Genome Biol.*, 6(7):224, 2005.
15. R. T. K. Lin, J. L.-T. Chiu, H.-J. Dai, M.-Y. Day, R. T.-H. Tsai, and W.-L. Hsu. Biological question answering with syntactic and semantic feature matching and an improved mean reciprocal ranking measurement. In *IRI*, pages 184–189, 2008.
16. X. Ling, J. Jiang, X. He, Q. Mei, C. Zhai, and B. Schatz. Generating gene summaries from biomedical literature: A study of semi-structured summarization. *Inf. Process. Manage.*, 43:1777–1791, November 2007.
17. D. Mollá and J. L. Vicedo. Question answering in restricted domains: An overview. *Comput. Linguist.*, 33(1):41–61, Mar. 2007.
18. A. Peñas and A. Rodrigo. A simple measure to assess non-response. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1415–1424, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
19. C. Perez-Iratxeta, M. Wjst, P. Bork, and M. A. Andrade. G2D: a tool for mining genes associated with disease. *BMC Genet.*, 6:45, 2005.
20. W. S. Richardson, M. C. Wilson, J. Nishikawa, and R. S. Hayward. The well-built clinical question: a key to evidence-based decisions. *ACP J. Club*, 123(3):A12–13, 1995.
21. R. Saetre, K. Yoshida, A. Yakushiji, Y. Miyao, Y. Matsubayashi, and T. Ohta. AKANE System: Protein-Protein Interaction Pairs in the BioCreAtIvE2 Challenge, PPI-IPS subtask. In L. Hirschman, M. Krallinger, and A. Valencia, editors, *Proceedings of the Second BioCreative Challenge Workshop*, 2007.
22. K. Sagae and J. Tsujii. Dependency parsing and domain adaptation with lr models and parser ensembles. In *Proceedings of the CoNLL 2007 Shared Task in the Joint Conferences on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07 shared task)*, pages 1044–1050, 2007. Prague, Czech Republic.
23. B. Settles. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192, Jul 2005.
24. Z. Shi, G. Melli, Y. Wang, Y. Liu, B. Gu, M. M. Kashani, A. Sarkar, and F. Popowich. 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*, CAI '07, pages 284–295, Berlin, Heidelberg, 2007. Springer-Verlag.
25. M. Simpson and D. Demner-Fushman. Biomedical text mining: A survey of recent progress. *Mining Text Data*, pages 465–517, 2012.
26. T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: a language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, 2005.

27. D. R. Swanson. Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspect. Biol. Med.*, 30(1):7–18, 1986.
28. K. Takahashi, A. Koike, and T. Takagi. Question answering system in biomedical domain. In *In Proceedings of the 15th International Conference on Genome Informatics*, 2004.
29. L. Toldo, S. Bhattacharya, and H. Gurulingappa. Automated identification of adverse events from case reports using machine learning. In *Proceedings XXIV Conference of the European Federation for Medical Informatics. Workshop on Computational Methods in Pharmacovigilance*, Pisa, Italy, 26-29 August 2012.
30. L. Toldo and A. Scheer. Finding patient visits in emr using luxid. In *The 20th Text REtrieval Conference (TREC 2011) Proceedings.* NIST Special Publication SP 500-295, 2011.
31. K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
32. E. M. Voorhees. The trec question answering track. *Nat. Lang. Eng.*, 7(4):361–378, Dec. 2001.
33. H. Yu and Y. G. Cao. Automatically extracting information needs from Ad Hoc clinical questions. *AMIA Annu Symp Proc*, pages 96–100, 2008.
34. H. Yu, M. Lee, D. Kaufman, J. Ely, J. A. Osheroff, G. Hripcsak, and J. Cimino. Development, implementation, and a cognitive evaluation of a definitional question answering system for physicians. *J Biomed Inform*, 40(3):236–251, Jun 2007.
35. P. Zweigenbaum. Question answering in biomedicine. In *In Proceedings Of The 10th Conference Of The European Chapter Of The Association For Computational Linguistics*, 2003.