Concept Recognition, Information Retrieval, and Machine Learning in Genomics Question-Answering

J. Gregory	Caporaso	William A.	Baumgartner, Jr.	Hyunmin Kim
Zhiyong Lu	Helen L. Je	ohnson	Olga Medvedeva	Anna Lindemann
Lynne M. Fox	Elizabeth K	. White	K. Bretonnel Cohen	Lawrence Hunter

larry.hunter@uchsc.edu Center for Computational Pharmacology University of Colorado Health Sciences Center

Abstract

TREC Genomics 2006 presented a genomics question-answering challenge with questions on twenty-seven topics, and a corpus of 162,259 full-text biomedical journal articles from which to derive answers. Questions were formulated from actual information needs of biomedical researchers, and performance was based on human evaluation of the answers. The University of Colorado approach to this task involved three key components: semantic analysis, document zoning, and a promiscuous retrieval approach followed by pruning by classifiers trained to identify near-misses.

We began by parsing the document HTML, splitting it into paragraph-length passages and classifying each passage with respect to a model of the sections (zones) of scientific publications. We filtered out certain sections, and built a search index for these passages using the Lemur system. Next, for each query, we semi-automatically created a set of expansions using ontological resources, including MeSH and the Gene Ontology. This expansion included not only synonyms, but terms related to concepts that were both more specific and (in some cases) more general than the query. We searched the passage collection for these expanded queries using the Indri search engine from the Lemur package, with pseudo-relevance feedback. We also tried expanding the retrieved passages by adding passages that had a small cosine distance to the initial retrievals in an LSA-defined vector space. Our final step was to filter this expanded retrieval set with document classifiers whose input features included word stems and recognized concepts. Three separate runs were constructed using varying components of the above set, allowing us to explore the utility of each. The system produced the best result for at least one query in each of the three evaluations (document, passage and aspect diversity).

Introduction

The University of Colorado approach to the 2006 TREC Genomics involved three key components: semantic analysis, document zoning, and a promiscuous retrieval style followed by pruning by classifiers trained to identify near-misses. Each of these components made a significant contribution to our results.

Semantic Analysis

Our approach to QA (and NLP generally) is to develop methods for detecting expression of particular conceptual or ontological entities, regardless of how they are expressed in the text. We took several complementary approaches to recognition of such entities for this task. First, our query expansion step used ontological resources such as MeSH and the Gene Ontology [2] to add *conceptually related* terms to the query. Not all such terms were synonyms; the goal was to do a broad retrieval of all documents that mentioned any concept related to the query. Second, we generated conceptual features for input to the pruning discriminators that were trained to distinguish among relevant documents and near-misses. We used conceptual features for recognizing generally important molecular biological concepts with many forms of expression (e.g. mutations), and for certain elements of the aforementioned ontological resources. Finally, we experimented with the Latent Semantic Analysis [8, 9, 14] approach to generate additional 'semantic' features for the discriminators.

Document Zoning

One of our early observations in preparing for this task was that many irrelevant retrievals were coming from particular document sections. For example, it was common to retrieve a citation to another document because its title was related to the query. Although the collection is diverse with respect to the way articles are divided into sections, the scientific publication idiom allows us to define generic document zones that should be ignored for the purposes of this task. Not only should the 'literature cited' section be ignored, but tables of abbreviations, acknowledgments, glossaries, and many generic document zones should be as well. Due to the diversity of document formats and section naming conventions, substantial development efforts had to be devoted to document zoning (see below). It is interesting to note that focusing NLP efforts on document zoning demonstrated value in many NLP applications [12, 13, 18].

Promiscuity and Pruning

Our semantic query expansion methods are good for increasing recall, by retrieving potential responses that mention topics related to the query. However, this promiscuous query expansion also produces many documents that have only a tangential relationship to the query. What counts as a relevant response to a complex query like the ones in the Genomics task has to be defined in terms of the proper combinations of semantic contents. The presence of one or another semantic feature alone is not adequate to capture relevance; interactions among semantic features must be used. The sorts of interactions among semantic features that define relevance for any particular query can be quite complex, so we took an empirical approach to defining them. The semantic features of the documents retrieved in the promiscuous phase where used to train classifiers. We used human relevance judgments on a small sample of the training data to train the classifiers, hence our 'interactive' classification. (In one run, we omitted this step, putting us in the 'manual' class.)

Methods

The University of Colorado team approached this task by generating a candidate result set which was subsequently expanded (to improve recall) and then pruned (to improve precision). First, we converted questions into term-expanded queries. These queries were passed to the Indri search engine from the Lemur project [15], which was indexed on all paragraphs from the document collection. Query results were zone-filtered to remove results from the document sections we considered likely to be entirely irrelevant. (We would have removed the paragraphs from the index completely given time, but instead used query constraints to rule out certain document sections. The particular sections ruled out could therefore be adjusted experimentally.) Pseudo-relevance feedback [10] on the top five returned documents was employed to expand search results. The collection of documents was again expanded from a pool of zone-filtered documents using latent semantic analysis. In a final, false-positive-eliminating step, we used naive Bayesian classifiers [20] trained on humanlabeled data with features including word stems and detected semantic concepts [19, 5, 11].

We submitted a total of three runs which were generated using different components of our system. In the following sections we provide details on the individual components, and then describe how they were used to generate our results.

Query preprocessing

Our first step in generating results for each topic involved converting the question into a query to be passed to our search engine. We chose to treat the different question templates identically, and focus our query processing efforts on expanding key terms to include synonyms which were likely to be important for system recall.

Query expansion was performed using Online Mendelian Inheritance in Man (OMIM) [1] and Information Hyperlinked over Proteins (iHOP) [6] to expand gene names to synonym lists. Semi-automated review of Medical Subject Headings (MeSH) terms was performed to expand terms involving biological processes, anatomy, and diseases to synonym lists. For general terms we similarly used the Gene Ontology to add more specific terms (e.g. *liver development* was expanded to include *bile secretion* and *vitamin* A synthesis). We additionally consulted with a resident ontological engineer to identify related concepts and to expand to appropriate broader terms (e.g. expanding *cell growth* to include *proliferation* and *tumor*). Abbreviations which seemed likely to lead to many false positives (e.g. AD for Alzheimer's disease) were dropped from the expanded queries.

Document preprocessing

According to the task definition, retrieved text spans could not cross HTML paragraph tags. In the first step of our work, we therefore built an HTML parser, split each full-text document on paragraph tags, recorded the original text spans and document sections, and assigned a unique identifier to each *paragraph*. Subsequent steps operated on the paragraph level, treating each as an independent document.

During this process HTML tags were stripped from each of the paragraphs, and the containing document section was recorded for each paragraph. Document sections were extracted by recognizing section headings in the html document, and associating all paragraphs contained between two section headings with the preceding section heading. We attempted to ignore subheadings.

Concept recognition

We were interested in using semantic features both in indexing our search engine, and as potential features for machine-learning-based classifiers. We differentiate semantic features from surface-level features on the basis that surface-level features (e.g. words, word bigrams) represent language that are used to represent an idea, while semantic conceptual features represent an attempt to identify the underlying meaning being presented. For example, one semantic concept that we are interested in recognizing is point mutation. Given the sentence: "To understand the role of active site residues in the protein's function, we performed functional assays on the A42G, Y56F, and S57A variants", if simply using tokens as features, the fact that there is a reoccurring mention of protein point mutations cannot be represented. However, a simple regular-expression-based pattern match for identifying point mutations in text [7, 17], recognizes the same concept, a point mutation, three times.

We were interested in recognizing several different semantic feature types. First, we wanted to recognize mentions of point mutations, as described above. For these we used the regular expressions of [7, 17], and additional patterns which are under development by the Center for Computational Pharmacology [Caporaso, unpublished work]. Next, we recognized disease and organ names as concepts by matching strings of text to MeSH identifiers from MeSH's disease and organ branches. When multiple MeSH identifiers could be mapped to the same text string, the longest exact match was selected. (For example, the text string type 1 diabetes could be mapped to MeSH identifiers for Diabetes or Type 1 Diabetes. In such cases, we selected the latter.) We additionally tried to recognize the concepts protein transport and biological process regulation by normalizing from verbs which are indicative of these concepts.

Zone filtering and document section normalization

Paragraphs were disallowed from document sections considered likely to be irrelevant: the *References, Acknowledgments, Abbreviations, Disclosures, Grants, Glossary, Table of Contents,* and *Materials and Methods* sections. Review of training data suggested that hits from these sections would likely be false positives. We considered a section such as *Abbreviations,* for example, to be particularly dangerous because it was likely to contain many important terms, but would not be relevant to a query. Probably our most surprising decision, to ignore *Material and Methods* sections, was based on the observation that these sections, which comprise a large portion of the corpus, often didn't contain information that was relevant to the queries presented.

We additionally included a length filtering step to disallow paragraphs which seemed to be the result of format errors in the original html files. In this step we eliminated paragraphs from our result sets which contained less than ten or greater than ten-thousand words. Paragraphs with less than ten words were often blocks of text contained within HTML paragraph tags, but which were not actually article paragraphs. Likewise, paragraphs containing more than ten-thousand words seemed to be the result of missing paragraph tags. A paragraph of greater than ten-thousand words would likely not be relevant to a query (in whole) and mistakenly returning one of these would be disastrous for system precision.

Successful zone-filtering of our corpus required normalization of section headings due to spelling and stylistic variations in the naming of sections. We compiled a list of the section headings we were able to extract ranked by their occurrence, and manually identified important normalization steps. In the most extreme example of spelling variations in the corpus, we found twelve variations in the spelling of *acknowledgments* which appeared at least twice. (Merriam-Webster's dictionary recognizes both *acknowledgments* and *acknowledgements* to be valid spellings.) Additionally, variations in section naming conventions were normalized to what we considered to be the section type. For example, we normalized sections referred to as *References*, *Citations*, *List of works cited*, and *Suggested reading* (among others) to a single type: *refer*, the Porter stemmer output of *references*, the most common of these variations.

We developed an algorithm for normalizing section headings based on manual analysis of a ranked listing of section headings occurring in the TREC corpus. The first step in this process involved converting the section heading to lowercase, replacing nonalphabetic characters with a single whitespace, removing leading and trailing whitespaces, and then applying the Porter stemmer [16] algorithm to all words. Next, two pattern matching steps were applied. First, an exact match was used to collapse the most common section headings to what we considered to be the section type. For example, in this stage we converted sections called Experimental procedures, Methods, and Study design to the type matmeth, on the basis that these sections generally discuss materials and methods used in the study. Last, we applied a non-exact pattern match step. This involved searching remaining unnormalized section headings for terms suggestive of specific section types. For example, if a section heading contained the term method, material, cell culture, or plasmid, we converted the section heading to *matmeth*. This inexact pattern matching allowed us to collapse many of the less commonly occurring section heading to underlying concepts.

Information Retrieval

We employed a probabilistic approach for the nonpromiscuous document retrieval portion of our system using the Indri search engine from the Lemur project[15]. A domain expert generated the queries used for retrieval by weighting the terms and synonyms in each of the expanded queries. We experimented with both the #combine and #band operators for combining query terms. Before submission to the search engine, each query was fitted with constraints to rule out document sections that were deemed irrelevant in our document zoning analysis. Porter stemming and stop word removal were utilized during both indexing and querying. Conceptual features were integrated into the index, although our final queries did not utilize these. The top five documents were used for pseudo-relevance feedback and we allowed up to one thousand documents to be returned from each Indri query.

Latent semantic analysis

To expand our result sets beyond what was achieved with Indri, query expansion, and pseudo-relevance feedback, we experimented with the sparseSVD (las2) algorithm), a C++ wrapper of the original SVD-PACKC package [4], to perform latent semantic analysis (LSA). We ran sparseSVD over the 5,350,887 paragraphs of our zone-filtered corpus, beginning with a total of 86,118 word stems (i.e. a roughly 5million-paragraph x 86-thousand-word matrix). The word stems were reduced by SSVD to 200 features, which resulted in 5,350,887 length two-hundred document vectors. We used these vectors to expand our results by finding documents similar to those output by Indri which were not already in the result set. Similar documents were identified by computing the cosine of each Indri-produced document vector against all other document vectors, following the document similarity used in [3], and returning the documents which yielded a cosine value greater than 0.6.

Relevance classification

In our final processing step, we applied naive Bayesian classifiers to retrieved paragraphs in an effort to eliminate false positives. Our classifiers were trained on a per-topic basis from human relevance judgements on selected paragraphs from the corpus.

Training corpora were developed for each topic by manually annotating positive and negative passages from results of simple Indri or grep queries. A group of four relevance judges reviewed possible answers for relevancy to be used for training. The judges were each assigned answer sets for six or seven topics. Judges reviewed at least 20 answers for each topic and marked the answer "not relevant," "definitely relevant," or "possibly relevant." Judges also noted the exact text span that included the answer, terms which might be useful in refining the initial Indri queries, and document sections which yielded high numbers of false positive results. Query expansion and zone-filtering were adjusted based on this information. Due to very limited annotation resources, these training sets generally contained only around 100 passages, with size varying with the number of query results.

During classifier optimization, we experimented with varied features types to identify those best suited to the task at hand. Feature types included features such as stemmed words and bigrams, and the various conceptual feature discussed above. We additionally compared performance of naive Bayes and Support Vector Machine classifiers, optimizing parameters for both. We found naive Bayes with kernel estimation to be best for this task.

Our best Weka [20] classifiers were constructed using word stems and DMAP-like concept matches as features. Feature selection was applied in two steps. First we removed stems which appeared in less than ten documents or more than fifty percent of the documents in the zone-filtered corpus, on the basis that these features would be too common or uncommon to be useful. Next, we calculated information gain (IG) for all remaining features and excluded features with information gain scores less than 0.001. (Most or all of the features eliminated by our first feature selection step would have been eliminated by IG selection, but pruning common and uncommon stems first greatly reduced IG compute time.) We used Weka's implementation of information gain, and optimization was performed to determine the optimal cut-off threshold using 10-fold cross validation of training data.

One classifier was constructed per topic, and these were applied to remove false positives before generating our submissions.

Results

We submitted three total runs, each of which included different components of our system. Our first run, uchsc1, utilized our full system. Expanded queries were sent to Indri and expanded with latent semantic analysis. LSA results were filtered with naive Bayes classifiers on a per-topic basis. In our second run, uchsc2, a looser set of queries were sent to Indri, but not expanded with LSA. Instead, the Indri results were pruned by the classifiers, and results submitted. In our last run, uchsc3, our first (strict) set of expanded queries were passed to Indri, and and those results were used without pruning by the classifiers. We categorized our first two runs as interactive (due to their reliance on manual annotation of training data for the pruning classifiers), and our third run, which did not use these classifiers, as manual. All runs were zone-filtered.

Table 1: **Performance compared to means.** Our first two runs, uchsc1 and uchsc2, were interactive runs. Our third run, uchsc3, was a manual run. All of our runs out-performed the means for each of the three performance metrics.

Run	Aspect	Document	Passage
	MAP	MAP	MAP
uchsc1	0.250	0.406	0.055
uchsc2	0.247	0.419	0.056
interactive			
mean	0.193	0.326	0.044
uchsc3	0.247	0.404	0.054
manual			
mean	0.132	0.277	0.027

Our systems performed well, with each of our runs scoring above mean on all three mean average precision (MAP) metrics. (Table 1) Each of our three submissions achieved the maximum of each of the three MAP scores for at least one topic. An interesting feature of our system performances is that we do well in topics where the median scores were low. (Figure 1) A possible explanation for this result is that in topics where few groups were able to identify relevant passages our systems were far more sensitive, essentially allowing us to set the gold standard. For example, in topic 170 we achieved perfect document MAP, returning passages from 806 different documents, while the median was 7 percent. For our system to achieve perfect document MAP (in a nontrivial situation) it seems likely that most of the relevant documents must have come from our submission, and were missed in submissions of other groups. We achieved the highest aspect diversity MAP for topic 170, the highest document MAP for topics 164, 170, 177, and 184, and the highest passage MAP for topics 161, 164, 170, and 174. (These comparisons are made against all runs classified as manual or interactive.) In situations where one of our runs outperformed the other two it was usually our second run (uchsc2) which did the best. However, in most cases, our three runs performed similarly.

Discussion

Section name normalization

Inherent ambiguity in section headings makes their normalization difficult. For example, it is often unclear what information differences exist between Results, Conclusions, and Discussion sections. In the TREC corpus, in addition to finding instances of sections named each of these terms, we found instances of all three pairwise combinations of these terms (e.g. Results and Discussion), and sections named some variation of all three terms combined (e.g. *Results*, Discussion, and Conclusions). To avert this issue, we converted all section headings containing any of the (stemmed) terms results, discussions, or conclusions to the single type resdisconc. However it is not clear that this is the best approach, or that a similar approach for other cases would be best. For example, is there enough difference in *Background* sections versus *Introduction* sections to warrant these being normalized to different types? For the purposes of TREC we considered these to be synonymous, but we feel that this could be argued either way.

Our work on normalization of section headings for TREC is crude, however we think that an accurate and general tool for zoning biomedical documents could be very useful in biomedical language processing. Our lab is currently exploring techniques for achieving this.

Utility of latent semantic analysis

The utility of LSA is not apparent when comparing mean average precision scores between runs, but in-



Figure 1: **Performance in passage, document, and aspect diversity mean average precision.** For (A) passage, (B) document, and (C) aspect diversity mean average precision, we present our best run compared with the median and best scores for the interactive/manual runs. Topics are sorted by median to illustrate that we often performed well in topics where the median performance was lowest. While we present our best run for each topic it should be noted that in most topics our three runs achieved equal performances. Also note that because the topics (x-axes) are sorted by median, and for each MAP that sort differs, the columns are not directly comparable between graphs.

Topic	uchsc1	uchsc2	uchsc3	Passage MAP	Document MAP	Aspect MAP
	(#d/#p)	(#d/#p)	(#d/#p)	(uchsc1/2/3)	(uchsc1/2/3)	(uchsc1/2/3)
160	341/802	272/830	222/682	0.05/0.05/0.05	0.77/0.79/0.77	0.16/0.16/0.16
161	967/1000	84/108	25/29	0.09/0.10/0.09	0.66/0.73/0.66	0.60/0.60/0.60
162	439/478	24/53	12/37	0.03/0.03/0.03	0.29/0.29/0.29	0.17/0.17/0.17
163	48/1000	26/1000	16/958	0.03/0.03/0.03	0.46/0.46/0.46	0.07/0.07/0.07
164	48/50	26/31	16/17	0.26/0.26/0.26	0.69/0.69/0.69	0.54/0.54/0.54
165	3/3	4/4	3/3	0.05/0.05/0.05	0.17/0.17/0.17	0.48/0.48/0.48
166	8/14	8/14	8/14	0.00/0.00/0.00	0.09/0.09/0.09	0.07/0.07/0.07
167	132/404	169/481	132/404	0.16/0.16/0.16	0.67/0.69/0.67	0.32/0.33/0.32
168	301/633	172/616	65/394	0.10/0.10/0.10	0.84/0.85/0.84	0.27/0.27/0.27
169	681/1000	678/1000	674/993	0.02/0.02/0.02	0.07/0.07/0.07	0.05/0.05/0.05
170	806/850	249/384	8/41	0.11/0.11/0.11	1.00/1.00/1.00	0.83/0.83/0.83
171	36/36	186/204	1/1	0.01/0.01/0.01	0.03/0.03/0.03	0.38/0.38/0.38
172	691/1000	561/880	536/841	0.00/0.00/0.00	0.21/0.22/0.21	0.01/0.01/0.01
173	30/33	98/118	1/4	no results	no results	no results
174	306/330	249/548	50/74	0.16/0.17/0.16	0.46/0.50/0.46	0.85/0.85/0.85
175	22/23	22/25	7/8	0.05/0.05/0.05	0.37/0.37/0.37	0.39/0.39/0.39
176	925/1000	286/345	10/26	0.01/0.01/0.01	0.13/0.18/0.13	0.06/0.06/0.06
177	3/4	34/41	3/4	0.08/0.08/0.08	1.00/1.00/1.00	0.58/0.58/0.58
178	474/976	430/1000	377/874	0.00/0.00/0.00	0.02/0.02/0.02	0.01/0.01/0.01
179	313/339	112/155	24/34	0.00/0.00/0.00	0.02/0.02/0.02	0.03/0.03/0.03
180	564/577	61/82	11/13	no results	no results	no results
181	309/486	396/727	211/386	0.11/0.12/0.11	0.62/0.70/0.62	0.12/0.13/0.12
182	415/1000	392/1000	372/950	0.00/0.00/0.00	0.19/0.19/0.19	0.01/0.01/0.01
183	0/0	0/0	0/0	0.00/0.00/0.00	0.00/0.00/0.00	0.00/0.00/0.00
184	200/233	18/47	6/33	0.02/0.02/0.02	0.83/0.83/0.83	0.14/0.14/0.14
185	11/11	26/29	4/4	0.02/0.02/0.02	0.21/0.21/0.15	0.20/0.13/0.13
186	555/893	453/831	380/709	0.03/0.03/0.03	0.35/0.36/0.35	0.11/0.11/0.11
187	128/160	117/174	14/36	0.02/0.02/0.02	0.44/0.44/0.44	0.03/0.03/0.03

Table 2: Comparison of UCHSC runs. For our three runs we present the number of documents (#d) and the number of passage (#p) returned, and the passage, document, and aspect MAP achieved by each.

spection of our submissions illustrates that its application often greatly increases the number of retrieved results. For example, in topic 170, applying LSA increases the number of passages we return from 384 to 850, and the number of documents we return from 249 to 806. (Table 2) (The most direct comparison is between runs uchsc1 and uchsc2, although in addition to LSA not being applied in uchsc2, Indri queries were slightly further expanded than in uchsc1. The expanded queries increased the number of hits from the search engine, so increases in the number of results due to LSA are understated.) Since the performance metrics used to judge submissions in this task do not highlight recall it is difficult for us to gauge the utility of LSA – without manually reviewing the results we cannot determine the relevance of the additional documents. However, based our observations in TREC, we expect that LSA would likely be beneficial in similar tasks where recall is important.

Semantics and inference

We thought it interesting to note that while the goldstandard annotators were instructed not to make any inferences when judging relevance, our best results required use of semantically related but not synonymous terms in our query expansion, clearly a sort of inference. The value of concept recognizers in the pruning step can also be seen as a demonstration of the value of (computational) inference even in this (human) inference-constrained task.

Acknowledgments

We would like to thank Xiangyu Jin for helpful discussion regarding probabilistic information retrieval.

References

- Online Mendelian Inheritance in Man, OMIM (TM)., July 2006.
- [2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A.

Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

- [3] J. Becker and D. Kuropka. Topic-based vector space model. In Proceedings of the 6th International Conference on Business Information Systems, pages 7–12, Colorado Springs, July 2003.
- [4] M. W. Berry. Large-scale sparse singular value computations. The International Journal of Supercomputer Applications, 6(1):13–49, Spring 1992.
- [5] Will Fitzgerald. Building embedded conceptual parsers. PhD thesis, Northwestern University, 1995.
- [6] R. Hoffmann and A. Valencia. A gene network for navigating the literature. *Nat Genet*, 36(7), July 2004.
- [7] Florence Horn, Anthony L. Lau, and Fred E. Cohen. Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors. *Bioinformatics*, 20(4):557–568, Mar 2004.
- [8] Alvarez E. Lacalle, B. Dorow, J. P. Eckmann, and E. Moses. Hierarchical structures induce long-range dynamical correlations in written texts. *PNAS*, 103(21):7956–7961, 2006.
- [9] T. K. Landauer, D. Laham, and M. Derr. From paragraph to graph: latent semantic analysis for information visualization. *Proc Natl Acad Sci* USA, 101 Suppl 1:5214–5219, April 2004.
- [10] Victor Lavrenko and Bruce W. Croft. Relevance based language models. In SIGIR '01: Proceedings of the 24th annual international ACM SI-GIR conference on Research and development in information retrieval, pages 120–127, New York, NY, USA, 2001. ACM Press.

- [11] Charles E. Martin. Direct memory access parsing. PhD thesis, Yale University, 1990.
- [12] Y. Mizuta and N. Collier. Zone identification in biology articles as a basis for information extraction. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, pages 29–35, 2004.
- [13] Tony Mullen, Yoko Mizuta, and Nigel Collier. A baseline feature set for learning rhetorical zones using full articles in the biomedical domain. SIGKDD Explor. Newsl., 7(1):52–58, June 2005.
- [14] David Newman, Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Analyzing Entities and Topics in News Articles Using Statistical Topic Models, volume 3975. January 2006.
- [15] Paul Ogilvie and James P. Callan. Experiments using the lemur toolkit. In *Text REtrieval Conference*, 2001.
- [16] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [17] Dietrich Rebholz-Schuhmann, Stephane Marcel, Sylvie Albert, Ralf Tolle, Georg Casari, and Harald Kirsch. Automatic extraction of mutations from Medline and cross-validation with OMIM. *Nucl. Acids Res.*, 32(1):135–142, January 2004.
- [18] Yizhar Regev, Michal Finkelstein-Landau, and Ronen Feldman. Rule-based extraction of experimental evidence in the biomedical domain: the kdd cup 2002 (task 1). SIGKDD Explor. Newsl., 4(2):90–92, December 2002.
- [19] Christopher K. Riesbeck. From conceptual analyzer to Direct Memory Access Parsing: an overview., chapter 8. Ellis Horwood Limited, 1986.
- [20] Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition, 2005.