Unsupervised Method for Automatic Construction of a Disease Dictionary from a Large Free Text Collection

Rong Xu[1], Kaustubh Supekar[1], Alex Morgan[1], Amar Das[1], Alan Garber[2]
[1]Center for Biomedical Informatics Research, [2]Center for Primary Care and Outcomes Research
Stanford University School of Medicine, Stanford, CA 94305, USA
xurong@stanford.edu

Abstract

Concept specific lexicons (e.g. diseases, drugs, anatomy) are a critical source of background knowledge for many medical language-processing systems. However, the rapid pace of biomedical research and the lack of constraints on usage ensure that such dictionaries are incomplete. Focusing on disease terminology, we have developed an automated, unsupervised, iterative pattern learning approach for constructing a comprehensive medical dictionary of disease terms from randomized clinical trial (RCT) abstracts, and we compared different ranking methods for automatically extracting contextual patterns and concept terms. When used to identify disease concepts from 100 randomly chosen, manually annotated clinical abstracts, our disease dictionary shows significant performance improvement (F1 increased by 35-88%) over available, manually created disease terminologies.

1 Introduction

Dictionary based natural language processing systems have been successful in recognizing biomedical concepts from free text. For example, the MetaMap program is used to map biomedical text to concepts from UMLS Metathesaurus[2]. It identifies various forms of UMLS concepts in text and returns them in a ranked list in a five-step process, identifying simple NPs, generating variants of each phrase, finding matched phrases, assigning scores to matched phrases by comparing them with the input and composing mappings. However, its performance largely depends on the quality of the underlying UMLS Metathesaurus and the associated Specialist Lexicon. A recent study has shown that, of the disease concepts identified by human subjects, more than 40% were not in UMLS[8].

Disease concepts are of core importance in medical text processing, but their terminology is highly dynamic. Individual authors may choose to represent the same disease many different ways. New diseases and conditions are also constantly emerging, such as SARS and avian influenza. Even advances in diagnostics and treatments can give rise to new disease modifiers such as those in 'HER2 overexpressing breast cancer' or 'Dexamethasone-responsive hypertension'. Clearly we need to develop techniques to deal with this dynamic terminology landscape.

Large quantities of biological text are available in Medline's collection of Randomized Clinical Trial (RCT) reports; over 500,000 RCT abstracts are available. RCT reports are a critical resource for information about diseases, their treatments, and treatment efficacy. These reports have the advantage of being highly redundant (a disease is often reported in multiple RCT abstracts), medically related, coherent in writing style and implicitly or explicitly structured, precise, trustworthy and freely available.

We have developed and evaluated an automated, unsupervised, iterative pattern learning approach for constructing a comprehensive medical dictionary of diseases from RCT abstracts. The algorithm starts with a seed pattern P_0, which represents typical written text about diseases. The program loops over a procedure, which starts by acquiring instances of diseases by matching the seed pattern in the parse tree of the sentences in RCT abstract, and discovers new patterns from the extracted diseases. The process is stopped when it reaches a fixed number of iterations. Diseases and patterns are assigned confidence scores before they are stored in a database. Our approach is inspired by the framework adopted in several bootstrapping systems in learning instances of concepts[1,3,4,5]. These approaches are based on a set of surface patterns introduced by Hearst[6], which are matched to the text collection and used to find instance-concept relations. A similar system is that of Snow and colleagues[9], which integrates syntactic dependency structure into pattern representation and has been applied to the task of learning instance-of relations or isa-relations.

All such systems suffer from the inevitable problem of spurious patterns and instances introduced in the iterative process. We have compared three different pattern ranking and three different extracted instance ranking approaches to address this issue.

2 Data and Methods

2.1 Data

421,471 RCT abstracts published in MEDLINE from 1965 to 2007 were parsed into 3,982,236 sentences. Each sentence was lexically parsed to generate a parse tree using the Stanford Parser[7]. The Stanford Parser is an unlexicalized natural language parser, trained on a non-medical document collection (Wall Street Journal). We used the publicly available information retrieval library, Lucene[1], to create an index on sentences and their corresponding parse trees.

2.2 Disease Extraction and Pattern Discovery

The pseudo code below describes the bootstrapping algorithm used in learning instances of disease and their associated text patterns. The algorithm starts with a seed pattern p_0, which represents a typical way of writing about diseases. For example, the seed pattern we used was "patients with NP" (NP: noun phrase). The program repeats a match procedure, which starts by acquiring instances of diseases by matching the seed pattern in the parse tree, and discover new pattern from the extracted diseases. In the disease extraction step, patterns are used as search queries to the local search engine. The parse trees with given patterns are retrieved and noun phrases (instances of diseases) following the pattern are matched from the parse trees. In the pattern discovery stage, diseases extracted from the previous iteration are used as search queries to the local search engine. Corresponding sentences with the diseases are retrieved and the bigrams (two words) in front of disease names are extracted as patterns.

```
initialize pattern_list, name_list
name_list_1 = extract_names(seed_pattern)
                    using  a seed pattern like "patients with",  this se-
                  lects the noun phrases to the right, returning terms
                   like "active rheumatoid arthritis" and "mild hyper-
                                                              tension"
for each name in name_list{
  patterns = extract_patterns(name)
                     returns patterns which are the two left tokens of
                       the matched noun phrase, bi-grams like "treat-
                         ment of", "diagnosis of", and "suffering from"
  append(pattern_list, patterns)
}
apply_ranks(pattern_lists, pattern_score_function)
                                scores and stores  the returned patterns
for each pattern in pattern_list {
  names = extract_names(pattern)
                     extracts  the noun phrases (of any length) which
                                          occur to the right of each patterns
  append(name_list, names)
}
apply_ranks(name_list, name_score_function)
                                scores and stores the final list of names
```

2.3 Pattern Ranking

Since the initial seed pattern is known to be good, the new pattern was scored on how similar its output (disease associated with the pattern) is to the output of the initial seed pattern. Intuitively, a reliable pattern is one that is both highly precise and general (high recall). Using the output diseases from the seed pattern (p_0) as a comparison, we developed specificity biased, sensitivity biased, and balanced algorithms to rank patterns. We define *diseases(p)* to be the set of diseases matched by pattern p, and the intersection [*diseases(p)* ∩ *diseases(p_0)*] as the set of diseases matched by both pattern p and p_0.

1. Specificity biased rank:

$$\text{Score1(p)} = \frac{\text{diseases(p)} \cap \text{diseases}(p_0)}{\text{diseases(p)}}$$

The specificity biased ranking method will favor patterns which hold for a few instances (diseases), but it may be too specific and may not be associated with any other disease. For example, complex patterns such as "aetiologically unclarified" precisely extract only one disease of interest (ARD).

2. Sensitivity biased rank:

$$\text{Score2(p)} = \frac{\text{diseases(p)} \cap \text{diseases}(p_0)}{\text{diseases}(p_0)}$$

The sensitivity biased ranking method will favor a general pattern, but it may be too general, which can introduce too much noise. For example, patterns containing only stop words essentially match every sentence in the data collection and thus will have a very high sensitivity biased rank.

3. Balanced rank:

$$\text{Score3(p)} = \frac{2 \cdot \text{Score1(p)} \cdot \text{Score2(p)}}{\text{Score1(p)} + \text{Score2(p)}}$$

A combination of the specificity biased and the sensitivity biased evaluation methods is the balanced ranking method, which takes into account not only the pattern specificity, but also the pattern generality (sensitivity). This method will favor general patterns while penalizing patterns which just hold for a few instances.

2.4 Disease Ranking

A reliable disease instance is one that is associated with a reliable pattern many times. We experimented with 3 ranking algorithms:

1. Abundance-based (document frequency) rank: A disease instance *(d)* that is obtained from multiple, distinct documents is more likely to be a real disease concept when compared with the one that appeared only

[1] http://lucene.apache.org/

once in the corpora. Also, since the documents in which the diseases appear are, in general, independently authored, the confidence of disease extraction increases with the number of supporting documents. We define *ScoreA(d)* as number of documents where a disease name, (*d)* appears in the RCT abstracts.

2. Pattern-based rank: A disease instance obtained from multiple patterns is more likely to be a real disease concept when compared with the one that was obtained by a single pattern. Ranked by the number of patterns that generated the disease *(d)*, score of those patterns, and the number of times that disease is associated with each of those patterns (Count (p,d)).

$$\text{ScoreB(d)} = \sum \log(\text{Score3(p)} \cdot \text{Count(p,d)})$$

3. Best-pattern-based rank: A disease instance obtained from highly ranked pattern is more likely to be a real disease concept when compared with the one that was obtained from a poorly ranked pattern. First the patterns are ranked by the best pattern *(p_b)* that generated the disease *(d)* and then further ties are broken by the number of times the disease is associated with that pattern (*Count(p,d*)) to provide *ScoreC(d).*

2.5 Evaluation

For evaluation and comparison purposes, we extracted the disease names from eight widely used sources: UMLS Metathesaurus, ClinicalTrials.gov[2], Cochrane Library[3], including Cochrane Review, Cochrane Economic Review and Cochrane Technology Assessment, OMIM[4], and PharmGKB[5]. Table 1 shows the eight data sources and the total number of distinct disease names (case sensitivity ignored) in each data source. We take these lists of disease names then to be manually compiled term lists, as each comes from a manually curated source.

Table 1: Number of disease terms in eight widely used disease sources

Data Source	Number of Diseases
UMLS Metathesaurus	482463
ClinicaTrial.gov	10372
Cochrane Review	5155
Cochrane Other Review	4542
Cochrane Economic Review	4149
Cochrane Technology Assessment	3080
OMIM	4056
PharmGKB	338
Combined	491949

[2] http://clinicaltrials.gov/
[3] http://www.thecochranelibrary.com/
[4] http://www.ncbi.nlm.nih.gov/sites/entrez?db=omim
[5] http://www.pharmgkb.org/

Evaluation of Stanford parser in identifying disease noun phrase boundary

Most disease names are noun phrases. We used the disease names from the eight sources as the gold standard to measure the accuracy of Stanford Parser in identifying disease boundary. With *NPcount(disease)* defined as number of times that Stanford Parser identifying the disease term as a noun phrase and *count(disease)* as number of times the disease term appears at all in the RCT abstracts.

$$\text{Accuracy} = \text{Average}\left(\frac{\text{NPcount(disease)}}{\text{count(disease)}}\right)$$

Evaluation of the extracted disease dictionary

We assessed the quality (precision and recall) of our dictionary by using it to identify disease concepts in 100 randomly selected RCT abstracts where disease names were manually identified by RX (first author). In addition, we also compared the performance of our dictionary with the eight manually curated disease sources.

3 Results

3.1 Evaluation of Stanford parser in identifying disease noun phrase boundary

Table 2 shows the accuracies of Stanford Parser in identifying noun phrase boundary for diseases from the eight data sources. The overall accuracy of Stanford Parser in identifying disease noun phrase boundaries was 0.95. Even though the Stanford Parser is trained on non-medical data, it is highly accurate in identifying disease noun phrase boundaries in the RCT abstracts.

Table 2: Accuracy of Stanford Parser in identifying disease noun phrase boundary

Data Source	Precision (%)
UMLS Metathesaurus	94.7
ClinicalTrial.gov	95.7
Cochrane Review	96.1
Cochrane Other Review	96.3
Cochrane Economic Review	96.1
Cochrane Technology Assessment	96.5
OMIM	94.7
PharmGKB	97.8
Combined	95.5

3.2 Evaluation of the extracted disease dictionary

Our derived dictionary consists of 1,922,283 potential disease names, each with an accompanying confidence score. We evaluated the quality of the dictionary by

using it to identify disease concepts in 100 randomly selected abstracts where disease names were manually annotated by one of the authors, RX. There was an average of four disease names per test abstract. Table 3 shows the precision, recall and F1 values using the best-pattern-based ranks of diseases (ScoreC) as the cutoff values. The precision, recall and F1 values at each cut-off were averaged across the 100 abstracts.

Table 3: Precision, recall and F1 at 7 cutoff values when tested on the 100 abstracts

Cutoff value	Precision	Recall	F1
48057 (Top 2.5%)	0.70	0.61	0.61
96114 (Top 5%)	*0.80*	*0.78*	*0.81*
144,171 (Top 7.5%)	0.72	0.79	0.72
199,228 (Top 10%)	0.59	0.81	0.64
240,285 (Top 12.5%)	0.58	0.81	0.64
293,250 (Top 15%)	0.58	0.82	0.64
336,399 (Top 17.5%)	0.58	0.82	0.63

Table 4 shows the precision, recall, and F1 values when disease names from the eight disease sources were used to identify disease names in the test dataset. As expected, for these manually created sources, the precision is high with values ranging from 0.68 to 1.0, while the recall is low with values ranging from 0.0 to 0.56. The recall of 0.0 for OMIM is due to the fact that all the diseases from OMIM are (sometimes very rare) genetic diseases and no disease names mentioned in the 100 test RCT abstracts had overlap with the OMIM vocabulary.

Table 4: Performance of eight disease sources in identifying disease names in the test abstracts

Data Source	Precision	Recall	F1
UMLS Metathesaurus	0.82	0.39	0.49
ClinicalTrial.gov	0.71	0.35	0.43
Cochrane Cochrane Review	0.68	0.54	0.57
Cochrane Other Review	0.70	0.56	0.60
Cochrane Economic Review	0.70	0.53	0.58
Cochrane Technology Assessment	0.67	0.51	0.55
OMIM	1.0	0.0	0.0
PharmGKB	1.0	0.32	0.48
Combined	0.44	0.68	0.50

The performance of our dictionary (F1 = 0.81, cutoff = top 5%) is a significant improvement over the eight widely used disease dictionaries (F1=0.0 to 0.6) and their combination (F1 = 0.5).

3.3 Disease Ranking

Table 5 shows the top 10 suggested disease names using "patients with" as the initial seed pattern. The rank of a disease instance is determined by the different (section 2.5) ranking methods: *abundance*, *pattern*, or *best-pattern* based ranking.

Table 5: Top 10 diseases with "patients with" as the seed pattern

Rank	Abundance based ranking	Pattern based ranking	Best pattern based ranking
1	patients	treatment	hypertension
2	treatment	patients	rheumatoid arthritis
3	the study	placebo	depression
4	this study	surgery	migraine
5	surgery	therapy	asthma
6	placebo	eat	duodenal ulcer
7	both groups	children	psoriasis
8	therapy	one	schizophrenia
9	baseline	time	breast cancer
10	children	baseline	obesity

None of the top 10 extracted phrases on the basis of *abundance* (ScoreA) or *pattern* (ScoreB) are actual disease names. These commonly used ranking methods will assign a high rank to common non-medical words. The *best-pattern* (ScoreC) based ranking method, as is evident from the table, correctly identifies diseases, mainly because it reduces the likelihood of selecting irrelevant patterns.

3.4 Pattern Ranking

Table 6 shows the top 10 patterns with "patients with" as the initial seed pattern.

Table 6: Top 10 patterns with "patients with" as seed pattern

Rank	Specificity Biased Rank	Sensitivity Biased Rank	Balanced Rank
1	*patients with*	*patients with*	*patients with*
2	cancer reduces	treatment of	treatment of
3	encoding under	treatment for	diagnosis of
4	carob as	diagnosis of	treatment for
5	optimally regulating	management of	management of
6	advancement throughout	incidence of	suffering from
7	with rehabilitation	women with	adults with
8	pulsatile blood	presence of	women with
9	neurons retain	patients had	incidence of
10	unoprostone or	rate of	patients without

The specificity-biased metric assigns high rank scores to very specific patterns such as "encoding under". The top 10 patterns based on the sensitivity-biased (Score2)

ranking and balanced (Score3) ranking are more disease specific.

When different seed patterns were used, most of the top 10 patterns were the same. For example, for the seed pattern "treatment of", 5 out of top 10 balanced-based ranked patterns were the same as those from seed pattern "patients with".

4 Discussion

We have demonstrated an automated, unsupervised, iterative pattern learning approach for bootstrapping construction of a disease lexicon with comprehensive coverage for text related to clinical trials. We also compared different pattern and extracted term ranking methods. We have shown that our automatically generated lexicon performs much better than lexicons constructed from manually compiled sources. Our approach is also potentially applicable to other concept categories such as drugs, treatments and gene terminologies which could then be combined to extract relationships between concepts. However, there is still significant space in which to seek improvement in increasing the coverage of our lexicon and the quality of our patterns.

Although useful in demonstrating the proof of concept and allowing us to examine different ranking methods, focusing on bigrams that preface noun-phrases limited the space of patterns that we could potentially examine. More complex patterns might involve longer n-grams, alternate word orderings (e.g. postfix patterns), using contrasting terminologies as filters, dependencies in the parse tree, or morphological features of the terms themselves.

For example, "*Necrotising sarcoid granulomatosis (NSG) is a rare disease diagnosed on the basis of pathological features" (PMID: 16264037)*. There is indeed a distinctive pattern following the disease name: "is a rare disease". Such patterns like "NP is a disease" or "NP is a rare disease" are valid patterns to identify diseases.

Another example is "*Treatment of the subjects with atorvastatin decreased the abundance of IL-12p35 mRNA in mononuclear cells (PMID 12492458)*". Simultaneously co-training multiple concept terminologies such as one of drugs, using existing terminologies, adding morphological features (e.g. "-oma" suffix of diseases) to filter key terms such as "atorvastatin" might all improve pattern quality.

Although our dictionary is not complete, as our corpus of literature increases, redundancy will increase the likelihood of a disease term being matched by a pattern. The rapid growth of biomedical knowledge and literature, which makes our automatically generated disease vocabulary necessary, can also act to increase its coverage over time.

All the data and code is available on request from the author.

Acknowledgments

RX is supported by training grant 5T15LM007033-22 from the U.S. National Library of Medicine.

References

1. Agichtein, E., and Gravano, L. 2000. Snowball: extracting relations from large plain-text collections. In Proceedings of the fifth ACM conference on Digital libraries (DL), 8594.
2. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp. 2001:17–21.
3. Brin, S. 1998. Extracting patterns and relations from the world wide web. In WebDB Workshop at 6th International Conference on Extending Database Technology.
4. Cimiano, P.; Handschuh, S.; and Staab, S. 2004. Towards the self-annotating web. In WWW '04: Proceedings of the 13th international conference on World Wide Web, 462– 471.
5. Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.;Shaked, T.; Soderland, S.; Weld, D.; and Yates, A. 2005.Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligence 65(1):91–134.
6. Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th conference on Computational linguistics.
7. Klein D and Manning CD. 2003. Accurate Unlexicalized Parsing. Proc of the 41st Meeting of the Association for Computational Linguistics, 2003; 423–30.
8. Pratt W, Yetisgen-Yildiz M :A Study of BiomedicalConcept Identification: MetaMap vs. People. Proc AMIA Symp 2003, 529-533.
9. Snow, R.; Jurafsky, D.; and Ng, A. 2005. Learning syntactic patterns for automatic hypernym discovery. In Proceedings of the 17th Conference on Advances in Neural Information Processing Systems (NIPS). MIT Press.