Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Biomedical Informatics 36 (2003) 145–158
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

A shallow parser based on closed-class words to capture relations in biomedical text

Gondy Leroy,[a,*,1] Hsinchun Chen,[a] and Jesse D. Martinez[b]

[a] *Management Information Systems, The University of Arizona, McClelland Hall, Room 430, 1130 E. Helen St., Tucson, AZ 85721, USA*
[b] *The Arizona Cancer Center, The University of Arizona, 1130 E. Helen St, Tucson, AZ 85721, USA*

Received 31 January 2003

Abstract

Natural language processing for biomedical text currently focuses mostly on entity and relation extraction. These entities and relations are usually pre-specified entities, e.g., proteins, and pre-specified relations, e.g., inhibit relations. A shallow parser that captures the relations between noun phrases automatically from free text has been developed and evaluated. It uses heuristics and a noun phraser to capture entities of interest in the text. Cascaded finite state automata structure the relations between individual entities. The automata are based on closed-class English words and model generic relations not limited to specific words. The parser also recognizes coordinating conjunctions and captures negation in text, a feature usually ignored by others. Three cancer researchers evaluated 330 relations extracted from 26 abstracts of interest to them. There were 296 relations correctly extracted from the abstracts resulting in 90% precision of the relations and an average of 11 correct relations per abstract.
© 2003 Elsevier Inc. All rights reserved.

Keywords: Natural language processing; Shallow parsing; Finite state automata; Biomedicine; Free text; Bottom-up parser; NLP

1. Introduction

Knowledge is largely disseminated in textual format and the texts are increasingly available online in electronic format. In the medical field, Medline is the main source of publications. It currently contains more than 12 million citations and is growing fast. However, the amount of available publications makes it hard for researchers to stay up-to-date.

Natural language processing (NLP) is a set of techniques that can help facilitate analysis, retrieval, and integration of textual and electronic information. NLP for medical texts has mainly been the focus of the medical informatics field. Bioinformatics, in contrast, has mostly focused on data processing, e.g., microarray analysis. During the last few years, NLP has also become important in bioinformatics and we agree with Maojo et al. [1] that both disciplines can and should learn from each other. In the following we describe natural language processing techniques for entity and relation extraction in medical informatics and bioinformatics. We then discuss our parser, which extracts relations between medical entities from biomedical text. This parser is the main component of a knowledge base for bioinformatics, Genescene, which we are currently developing. Genescene stores Medline abstracts relevant to several biomedical topics, e.g., AP-1, p53, yeast, together with the relations extracted from these abstracts. We describe here the details of the biomedical parser, which automatically extracts the relations.

2. Natural language processing for medical text

2.1. Entity extraction

When dealing with medical text, it is important to recognize different entities such as diseases, symptoms, and gene names. Two main approaches to recognize these entities exist. The first uses existing, manually created

* Corresponding author. Fax: 1-909-621-8564.
E-mail addresses: gondy.leroy@cgu.edu (G. Leroy), hchen@eller.arizona.edu (H. Chen), jmartinez@azcc.arizona.edu (J.D. Martinez).
[1] Current address: School of Information Science, Claremont Graduate University, 130 E. Ninth St., Claremont, CA 91711, USA.

1532-0464/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/S1532-0464(03)00039-X

knowledge sources containing lists of specific entities, such as disease names or human gene names. Words are compared against the entities in these sources and tagged with an appropriate tag. A second approach uses partial parsers to recognize entities in text. The two approaches can very well be combined, especially if there does not exist a single knowledge source that contains a sufficiently large and specific vocabulary to deal with all medical text.

2.1.1. Knowledge sources

The UMLS, developed by the National Library of Medicine [2] and available from http://umlsks.nlm.nih.gov, is a knowledge source useful for entity extraction from general medical text. It is updated yearly and currently consists of three components. Its Specialist Lexicon contains general English and specific medical terms and their syntactic and orthographic information. The Metathesaurus is a concept-based vocabulary in which each concept represents several unique terms. The concepts are an abstract representation of the words used in text. In addition to these concepts, the Metathesaurus contains a list of phrases that are represented by these concepts. For example, the concept "Genes, p53" is mapped to seven phrases, e.g., "P53 Tumor Suppressor," "ONCOGENE, P53." In addition, a concept has one or more semantic types, e.g., the word "RB1" belongs to the concept "RB1 Gene" with the semantic type "Gene or Genome." The Semantic Net links the semantic types by means of semantic relations.

Several researchers have evaluated the coverage of existing knowledge sources. However, since different mapping strategies are used, it is often difficult to compare the results. Hersh et al. [3] evaluated more than 200,000 documents and found that less than 40% of the words appeared in any of their six vocabularies, one of which was the Unified Medical Language System (UMLS). Aronson [4] developed a more comprehensive mapping technique, MetaMap, to map biomedical terms to the UMLS Metathesaurus. This mapping is not limited to string matches, but uses variant generation and knowledge intensive algorithms to choose between different candidates. Humphreys et al. [5] performed an extensive evaluation to estimate to what extent the UMLS can provide the vocabulary required by health information systems. Most of the terms submitted by their participants belonged to the field of individual healthcare. The authors found that 58% of all submitted terms had exact matches, 41% had related concepts, and only 1% of the terms were not found.

Specific biomedical knowledge sources are also available. The Human Genome Nomenclature (HUGO) [6], available from www.gene.ucl.ac.uk/nomenclature, contains a list of more than 15,000 currently approved human gene names and symbols and the names and symbols used prior to approval. Its purpose is to facilitate communication and electronic information retrieval of human genes. The Gene Ontology (GO), available from www.geneontology.org, contains information on genes and their products and is designed for use by both people and computers. The GO Consortium has several member organizations such as the Berkeley Drosophila Genome Project, FlyBase, and WormBase. By integrating different organisms, they aim to provide a controlled vocabulary for biomedical research useful for all eukaryotes [7,8]. The GENIA corpus [9] is a small, tagged, biomedical corpus available from http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/ and can be used to train entity extraction algorithms. There are many more specific biomedical databases such as Flybase (http://flybase.harvard.edu), *Caenorhabditis elegans* (http://elegans.swmed.edu), Online Mendelian Inheritance in Man (OMIM, http://www3.ncbi.nlm.nih.gov/Omim/), or SWISS-PROT (http://www.ebi.ac.uk/swissprot/).

2.1.2. Extraction algorithms

Partial parsers are very efficient for entity extraction because they focus only on those elements of interest. There exist several partial parsers for general English text. For example, Weischedel et al. [10] focused on tagging unknown words, core noun phrases, and verb frames. They found that, for example, noun phrases could be discovered in text with a 90% success rate when using only local information and a statistical partial parser. Hindle [11] developed a partial parser to discover non-fluencies, i.e., the corrections a speaker makes while speaking, in transcribed text. This deterministic parser edits the text where the non-fluencies occur. Church [12] developed a partial parser based on dynamic programming that assigns parts-of-speech in real time to be used with speech processing. The parser used local information only in a bottom-up fashion and assigns correct tags in more than 95% of the cases. Voutilainen and Padro [13] combined linguistic rules with statistical disambiguation to extract noun phrases from text. This hybrid approach achieved both high precision (97%) and recall (96%). McDonald developed a parser to extract 4-tuples consisting of person and company names, title, and events [14], which correctly identified 81% of these 4-tuples in Wall Street Journal articles.

There also exist partial parsers that focus on biomedical and medical entity recognition. The Arizona (AZ) Noun Phraser was originally developed as a general English noun phraser and later adapted to recognize *relevant* medical phrases [15]. The evaluation showed that recall of these relevant medical phrases was 52% with a precision of 36%. Hersh et al. [16] focused on general medical concepts and matched the medical text from radiology reports to UMLS Metathesaurus concepts. Their goal was to automatically add indexing terms to the reports. They discovered more than 60% of the indexing terms with 30% precision.

Entity recognition tends to improve when the algorithms are developed for more specific entities. Rindflesh et al. [17] looked at "binding terminology" and combined heuristics and the matching of extracted noun phrases with the UMLS Metathesaurus and NCBI's GenBank to discover binding terminology in Medline. They recalled 72% of the binding terms with 79% precision. Raychaudhuri et al. [18] assigned GO annotations to genes. They used a document classifier based on the maximum entropy principle to associate Medline abstracts with GO annotations. Then they annotated genes by combining and weighting all GO annotations associated with the abstracts discussing the genes. Their maximum entropy model achieved 72% accuracy. Kazama et al. [19] trained support vector machines on the GENIA corpus to assign words to 24 entity classes. Although the technique is very promising, the authors report that precision was too low for practical use. Fukuda et al. [20] used surface clues of strings to recognize materials names, e.g., proteins, with high precision (95%). Cohen et al. [21] developed four types of heuristics to match gene names found in Medline to their official name. The best results, 85% precision, were achieved with strict pattern matching. Hatzivassiloglou and Dubou'e [22] tested three machine learning approaches, native Bayesian, decision trees, and inductive rule learning, to distinguish between genes, proteins, and RNA in text. They achieved approximately 80% accuracy when testing non-ambiguous cases, i.e., terms containing disambiguating words that were not used for learning. When comparing against labels assigned separately by three experts, accuracy was approximately 70%. The pairwise agreement of their three experts was 77%, illustrating the complexity of the task. Krauthammer et al. [23] used BLAST, available at http://www.ncbi.nlm.nih.gov/BLAST/, to assign nucleotide sequences to words and so recognize gene and protein names in text. They compared their automated technique with a list of gene names compiled by experts. When comparing the combined set of full and partial matches of extracted names with the expert list, they achieved 79% recall of the genes and proteins with 71% precision. For names already available in the BLAST database, they achieved 95% full matches. Proux et al. [24] used cascaded finite state transducers to recognize gene names in sentences from the FlyBase set. After tuning, they could extract 94% of the gene names with 91% precision. Liu et al. [25] used different versions of a naïve Bayes and a Decision List Method to disambiguate 12 biomedical terms using the UMLS Metathesaurus. Their best classifier reached an overall accuracy of 97%.

2.2. Relation extraction

Besides recognizing medical entities, it is important that the relations between them are extracted from the text. Several different techniques exist. With co-occurrence based approaches, the entities are first extracted and the relations are based on the assumption that two entities in the same sentence or abstract are related. Negation in the text is not taken into account. Jenssen et al. [26] collected a set of almost 14,000 gene names from publicly available databases and used them to search Medline. Two genes were linked if they appeared in the same abstract; the relation received a higher weight if the gene pair appeared in multiple abstracts. For the pairs with high weights, five or more occurrences of the pair, the authors found that 71% of the gene pairs were indeed related.

Linguistic-based relation extraction usually employs time-efficient, shallow parsing techniques focusing on specific parts of the text and predefined, handpicked verbs and nouns. Rules are specifically developed to extract the surrounding words of these predefined terms and to format them as relations. As with the co-occurrence based approach, negation in sentences is usually ignored. Many techniques achieve high precision but low recall. This is no surprise since only a small number of relations between genes or proteins can be automatically captured when the exact terms (the genes or proteins) need to appear in the sentence. Blaschke and Valencia [27] found that only 25% of all existing protein interactions could be found in sentences in Medline abstracts.

Sekimizu et al. [28] collected the most frequently used verbs in a collection of abstracts and developed partial and shallow parsing techniques to find the verbs' subject and object. They estimated their precision at 73%. Thomas et al. [29] modified a preexisting parser based on cascaded finite state machines to fill templates with information on protein interactions for three verbs: interact with, associate with, bind to. They calculated recall and precision in four different manners for three samples of abstracts. Recall ranged from 24 to 63%, and precision from 60 to 81%. Pustejovsky et al. [30] targeted *inhibit* relations in the text and built finite state automata to recognize these relations. They achieved 91% precision and 59% recall on 56 abstracts. The PASTA system is a more comprehensive systems that extracts relations between proteins, species, and residues [31]. This system fills templates representing the relations between these three elements where appropriate. The authors achieved 82% precision and 84% recall for the recognition and classification of the terms, and 68% recall and 65% precision for the complete templates. GENIES [32] uses the MedLEE parser [33] to retrieve target structures from full-text articles. The authors report very high precision (96%) for relations between biological molecules found in full-text articles. They also use predefined verbs and templates for each, which are encoded in a set of rules.

MedLEE is probably the most advanced medical natural language processing system not part of

commercial for-profit software. It was originally developed for chest radiograph reports, has been expanded to cover several other domains, and is currently used in a clinical setting to automatically encode the information in both chest radiograph and mammogram reports [34]. It consists of five modules: a preprocessor to perform lexical lookup, a parser that identifies structures, a compositional regularizer to compose phrases from words, an encoder to map terms to codes, and a recovery component to take care of failed parses [34,35]. The relations MedLEE extracts are based on a semantic grammar. The parser starts from a controlled vocabulary and hundreds of grammar rules to recognize patterns.

The parser we are developing has a syntactic basis. All relations are processed without limiting in advance what type of content is to be captured. The advantage of our approach is that we can extract many different relations with a small, manageable number of rules. The advantage of an approach such as MedLEE's is that meta-knowledge of a relation, e.g., if a phrase indicates a body part, is available from the start. We will later attempt to use ontologies and vocabularies to tag the elements in our relations with this type of meta-information.

3. Parser development

3.1. Purpose

The parser is part of Genescene, a knowledge base we are building for biomedical researchers. It extracts relations from abstracts, which are stored in a document warehouse together with the original abstracts and all abstract meta-information. A parser relation can contain up to five elements: relation negation, left-hand side element (LHS), connector modifier, connector, and right-hand side element (RHS). For example, from the abstract title: "Regulation of E2F1 activity by acetylation," the following relation is extracted: "Acetylation (LHS)—regulates (connector)—E2F1 activity (RHS)." In some cases, a modifier (one or more adverbs) that adds information about the connector and negation of the relation are also available. More detailed descriptions are provided in the following section.

Most existing techniques are developed from a semantic perspective. They specify few elements or relations of interest, e.g., gene names or verbs, and build parsers that recognize patterns around these specific words. We started from a syntactic perspective and extract relations between all noun phrases regardless of their type. Comparable to others, we built our parser to look for certain patterns in the text; however, these patterns are based on English closed-class words, e.g., conjunctions and prepositions. This provides us with templates that are generic and do not depend on a prespecified medical vocabulary. Our goal is to extract relations in a very precise manner. The relations, however, are not limited to a few nouns and verbs. Our focus on precision is necessary because we later want to use these relations to visualize the content of texts or to perform text mining on the relations and researchers distrust software that is based on incorrect biological information.

3.2. Overview

Our parser consists of two modules. The semantic module captures the content of the abstracts, and the structure module consists of cascaded finite state automata (FSA) to provide structure for the content. Both modules are described in detail below.

3.2.1. Capturing content

There are five sequential processes used to capture the content of abstracts. Fig. 1 provides an overview. Abstracts are first cleansed by removing phrases referring to the publisher and copyright information. Text between parentheses is removed when not part of a biomedical term, but nothing is removed from for example, "H(2)O(2)." The abstracts are then split into sentences based on punctuation. We use a short list of heuristics to ensure that a sentence is not split incorrectly. The heuristics deal with phrases such as "$p < .01$," common English abbreviations such as "vs.," and biomedical-specific abbreviations such as "*Escherichia coli*."

Each sentence is submitted to the AZ Noun Phraser [15], which extracts medical nouns and noun phrases from the sentences. The settings of the AZ Noun Phraser were adjusted so that it does not extract com-

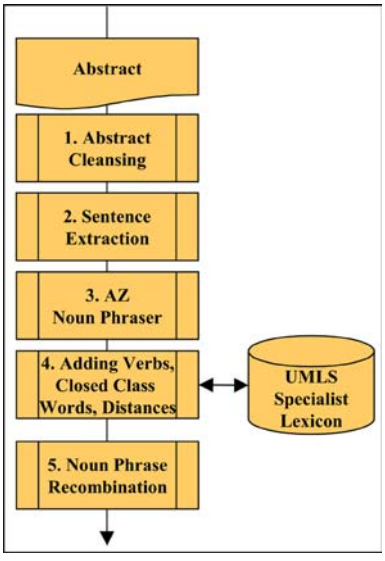


Fig. 1. Capturing semantics.

plete prepositional phrases, but instead splits them up in the constituent phrases. Nouns not recognized by the AZ Noun Phraser are added based on lexical lookup and a set of heuristics. For example, words that have a dash in the middle are considered nouns. This heuristic is based on the observation that authors use words such as "self-epitopes," or "TGF-β" in a relation such as "Generation of active TGF-β." A second set of heuristics is used to combine nouns together. For example "G1" was recognized based on lexical lookup and was combined where appropriate with "cell cycle" when both words appeared together in the text as "G1 cell cycle." As such, the parser is not limited to phrases or words that appear in controlled vocabularies. This approach illustrates a first difference between our parser and the semantic approach. For example, for MedLEE a knowledge engineer is trained to add terms to its lexicon [35]. Only terms that are part of the lexicon can become part of a relation.

Nouns and noun phrases are also checked to discover if they consist of nominalizations. We use the UMLS Specialist Lexicon for this purpose. When a nominalization is discovered, e.g., "activation," then both the underlying verb infinitive and the original nominalization are retained. Which one will be used in the relation is decided in a later phase.

Closed class words such as prepositions, negation, conjunctions, and punctuation are also tagged. Determiners and pronouns are ignored. Verbs and adverbs are recognized and added based on lexical lookup. The UMLS Specialist Lexicon is used for lookup. The number of ignored or unrecognized words in the sentence between extracted elements is then added as the "distance" between these elements. This distance measure will allow us to retain sufficient precision when combining the entities into relations.

3.2.2. Capturing structure

Overview. To capture the structure of a sentence, we use a shallow parser based on closed class English words. The closed classes' membership does not change and allows us to build very specific but semantically generic relation templates. We initially chose only prepositions and negation but later added conjunctions. Prepositions were chosen because they form the connections between different elements in a sentence. Prepositions are the heads of phrases [36] and we believe that these connections, in addition to verbs, are important in representing free text as binary relations. Prepositions indicate different types of relations between phrases, such as time or spatial relations [37], but can also be distinguished based on their operative class [38]: predicative and non-predicative prepositions. The first indicate a semantic relation; the prepositions are used to communicate information about an object, action or process, e.g., after, under. The second indicates a syntactic relation; the prepositions are used to indicate cases within a clause, e.g., from and with. However, prepositions can belong to different classifications depending on their use in a sentence.

We chose three prepositions (by, of, in) for several reasons. We wanted to use prepositions that occur frequently in the text and that led to interesting relation templates for the researchers. In addition, we felt it was interesting to test both prepositions with and without grammatical function. "By" and "of" have a grammatical function in the sentence and do not contribute much to the meaning. "By" is used very often to head complements in passive sentences, for example, in "Mdm2 is not increased by the Ala20 mutation." "Of" is one of the most highly grammaticized prepositions that allows a wide range of semantic relations between phrases [36] and is often used as a complement, such as for example in "the inhibition of the activity of the tumor suppressor protein p53." In contrast, "in" is usually a positive indication of location and it forms interesting relations when combined with the verb, for example in "Bcl-2 expression is inhibited in precancerous B cells." In addition, extracting relations from both active and passive sentences provided us with a test case where we could evaluate how suitable it would be to change relations from passive to active.

A problem encountered when using prepositions for text analysis is that of prepositional attachment. For example, in the sentence "He bought the shirt with pockets," the phrase "with pockets" goes with the noun "shirt." However, in the sentence "He washed the shirt with soap," the "with soap" goes with the verb "washed." Different approaches are used to disambiguate the attachment. Maximum entropy models [39], rule-based approaches [40], and several machine learning methods [41,42] have been used. However, we believe that for a specific and scientific domain, this problem will be less pronounced because authors try to communicate their message in an unambiguous manner. In addition, the same type of sentence structures is often used, making it probable that these structures will also have the same attachment.

We tested our parser initially with two prepositions [43] and have expanded it since then. The templates are currently based on three prepositions (by, of, in), two conjunctions (and, or), the comma, negation, and auxiliary or modal verbs to structure relations. One of our reasons for chosen these prepositions, as described above, was their frequency of appearance, see Table 1. Although "with" occurred more often than "by," we chose "by" because it is frequently combined with "of" and produces interesting relations. "To" was more often an infinitive marker than a preposition.

The parser recognizes most conjunctions, prepositions, determiners, and auxiliary verbs but only a subset is currently used in the templates. Table 2 provides an

Table 1
Prepositions in sample biomedical abstracts

	Total	%
Abstracts	500	
Sentences	6434	
Prepositions	16,411	100
Of	5416	33
In	3416	20
To	2145	13
With	1324	8
By	1126	7
For	918	6
On	452	3
41 Other prepositions	1614	10

Table 2
Closed class words recognized and utilized to extract relations

	Recognized	Utilized
Modal/auxiliary verbs	35	35
Conjunctions	52	2
Determiners	28	Ignored
Negations	26	26
Prepositions	63	3
Punctuation	8	1 (comma)

overview. Recognizing elements that are not yet utilized will facilitate future expansion of the parser.

Negation is a complex phenomenon in both spoken and written text. It can be easily detected when there are specific negation words present, e.g., not, neither, and never. This is both the case for *Not*-negation, e.g., not, n't, and *No*-negation, e.g., never, nobody. However, in many cases, these specific words are not presented, e.g., deny, fail, and lack. Words in this second category are called inherent negatives [44], i.e., they have an negative meaning but a positive form. An additional difficulty is that negation can be non-affixal or affixal. Examples of the non-affixal negation are the words no, none, and nothing; examples of the second affixal negation are word ending in -less, e.g., childless, or words starting with non-, e.g., non-committal. We are concerned with both not- and no-negation of the non-affixal type because these form a closed class [44] and all the components can be enumerated.

Few people dealing with medical or biomedical text report on capturing negation. A first interesting exception is the work by Chapman et al. [45] who captured negation in narrative medical reports. The authors developed regular expressions to look for negation patterns and could recall 88% of the patterns covered by the expressions with 68% precision. MedLEE's parse module deals with negation by treating negation as an atomic or leaf category in its parser's grammar [34,46]. This category is contained in several modifying grammar rules. Mutalik et al. [47] hypothesized and discovered that most negations in medical text are straightforward and can be captured with regular expressions. They achieved recall and precision of over 90% when parsing negations in their test set. Since the list of possible negations is limited, our parser uses a list containing for example "not," "neither," and "isn't" to recognize these elements. We do not treat verbs such as "inhibit" as negative instances of other verbs such as "activate."

Relation recognition. FSA represent the relation patterns of interest. An FSA consists of a set of nodes and the arcs that connect them. The nodes and arcs are organized as a directed graph. Fig. 2 shows an example of a simple automaton, with four nodes or states, that can recognize noun phrases. Moving from one state to the next is called a state transition. The start state is indicated by q_0. For example, the phrase "terrible disease" consists of an adjective and a noun and would be recognized by the following transitions: when the adjective is encountered, there will be a transition from state q_0 to state q_1 because the input is an adjective and so it the label on the arc between state q_0 and q_1. Then, when the noun is encountered, there will be another transition from state q_1 to state q_3, which is an end state or final state. End states are indicated with a double circle in the graph. Other phrases such as "very terrible disease" consisting of an adverb, adjective, and noun would also be recognized (order of states: 0, 2, 1, and 3). This simple automaton requires that each word has received a correct label. When an element is encountered that is not in the finite state automaton, e.g., a verb encountered when in state q_1, the automaton rejects the input, or it is said to end in a fail or sink state [48]. Sink states are usually not explicitly represented in the model but are used to describe the lack of possible advancement in the FSA.

FSA can be deterministic or non-deterministic. Deterministic FSA do not have decision points, i.e., states with arcs leading to different nodes for the same input. The FSA in Fig. 2 shows a deterministic FSA because at every state, only one arc can be followed for certain input. If, for example, state q_2 had two arcs labeled "adjective" that pointed to a different node, then this would have been a non-deterministic FSA. Finite state transducers are very closely related to FSA. The transducers differ from automata in that they have a pair of symbols on each arc: one input symbol and one output

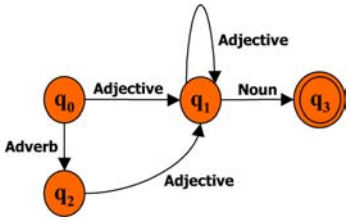
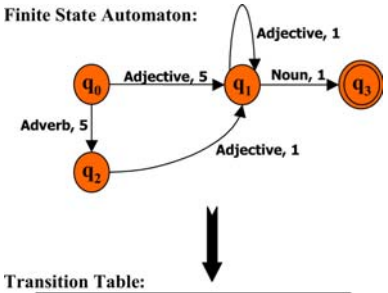


Fig. 2. Example finite state automaton.

symbol. The automata are used to recognize a sequence of elements, i.e., patterns; the transducers can transform these elements while transitioning from one state to the next. For formal definitions, we refer to Roche and Schabes [49] and Jurasfky and Martin [50].

FSA and transducers are commonly used in many different aspects of natural language processing, such as parts-of-speech tagging, parsing, or chunking. For example, Kokkinakis and Johansson-Kokkinakis [51] used cascaded FSA to parse Swedish and achieved 95% precision and 92% recall for full chunk parsing. Grefenstette [52] used a finite state approach to identify noun and verb groups and the syntactic relations between and within groups. Abney [53] described a chunker for English and German based on cascaded FSA that is almost as precise as the human evaluators. FSA are also successfully applied towards less common tasks. For example, Van Delden and Gomez [54] used FSA to determine the syntactic roles of commas; precision was well over 90%.

FSA are efficient and can deal with complex structures. Roche and Schabes [49] showed that a finite state version of a rule based part-of-speech tagger runs at much greater speed. Roche [55] showed that finite state transducers can be used to handle linguistically complex structures, demonstrating the efficiency of the automata structures in general. We used FSA to accept or reject patterns found in the input. When the FSA accepts a pattern, we store information about this pattern as a relation. The FSA used to accept patterns, i.e., recognize relations, are very similar to that in Fig. 2 with additional features such as can be seen in Fig. 3. In addition to a label on each arc that indicates what type of transition is allowed, our FSA also have a maximum distance on each arc. The maximum distance allowed between elements is used as an additional restriction to increase the precision of the relations. This distance is the number of steps required to get from one element to the next. For the example in Fig. 3, one step is allowed to get from the adjective "terrible" to the noun "disease" which means that there can be no intervening words. The first adjective of a noun phrase can be the fifth word in a sentence. This ensures that elements separated by unrecognized or unsuitable words are not part of a relation. Blascke and Valencia [27] also use the number of intervening words between protein names in a sentence and assigned a lower score to relations covering a larger distance. In our case, the distance is not used as part of a score but to decide if the FSA can advance to the next state or if a sink state is encountered. Another adaptation is that we use a short list of irrelevant phrases that lead to sink states. This ensures that we do not store relations such as "the aim of this study," and do not spend time processing such irrelevant relations. The directed graph is represented by a state transition diagram, as is also shown in Fig. 3.



State:	Input: Adj.	Adv.	Noun
q_0	1	2	*
q_1	1	*	3
q_2	1	*	*
q_3:	*	*	*

Fig. 3. Augmented finite state automaton and transition table.

During state transitions, the elements are recognized based on their tags, such as noun or verb, and the actual strings are temporarily stored. Each element is retained together with information about the state it fitted. For our first example, we store (state 1, terrible) and then (state 2, disease). If the FSA ends in a successful end state, these stored strings are permanently stored in relation format.

The parser is currently based on four cascaded FSA. Cascading FSA means combining them by adding arcs between the FSA such that transitions can be made from one FSA into the next one. To improve the speed of our parser, we made a few modifications in comparison to classical, cascaded FSA. For our cascade, the parser can progress past an end state in search of a later end state. This is done in a greedy manner, meaning that it passes successful end states until a sink state or the last possible end state is encountered. When the parser encounters a sink state, it backtracks to the last encountered end state. A further modification was that the four FSA share some states where the patterns are identical. This made the parsing more efficient: when one pattern cannot be successfully accepted, it is not necessary to start over and process the same states a second time for a different FSA.

Relation format. The patterns recognized by the FSA are later stored as relations in the database. They express binary relations between two noun phrases. Relations can contain up to five elements and require a minimum of two elements. The *left-hand side* (*LHS*) of a relation is often the active component and the *right-hand side* (*RHS*) of a relation is often the receiving component. The *connector* connects the LHS with the RHS and is often a verb. The relation can also be *negated* or augmented with a *modifier*. We will present examples in the following format: "negation: LHS—(modifier)

connector—RHS'' to facilitate readability. For example, from the sentence ''... Hsp90 has become a promising new drug target'' a relation is extracted as follows ''Hsp90 (LHS)—become (connector)—promising new drug target (RHS).'' From the sentence ''Thus hsp90 does not inhibit receptor function solely by steric interference; rather ...'' the following relation is extracted ''NOT(negation): Hsp90 (LHS)—inhibit (connector)—receptor function (RHS).'' Passive relations based on ''by'' are stored in active format. In some cases the relations are simpler, e.g., when the connector is only a preposition, e.g., the relation ''single cell clone—of—AK-5 cells.'' In other cases, the preposition ''in'' and the verb are combined, e.g., the relation ''NOT: RNA Expression—detect in—small intestine,'' or multiple adverbs are combined into one modifier.

BS-FSA. Fig. 4 provides an overview of the FSA for Basic Sentences (BS-FSA). This FSA models short basic sentences containing minimally two nouns or noun phrases and a verb. This pattern is often found in the title of an abstract, but can also be part of a longer sentence. On each arc in Fig. 4, there is a label to indicate the required input to advance and the maximum distance allowed. The start state is q_0. The parser progresses from state 0 (q_0) to state 1 (q_1) when it encounters a noun phrase that is not more than five words from the start of the sentence; it progresses to state 4 (q_4) when negation is encountered. The FSA requires a verb and a final noun phrase to lead to two possible success states (q_{19} and q_{20}). Modifiers, auxiliary verbs, and negation are optional to progress but are captured when encountered. For example, from the sentence ''Yet, E2F1 deficiency does not accelerate tumor growth,'' the following relation is extracted: ''NOT: E2F1 deficiency—accelerate—tumor growth.'' This FSA contains two sets of common states that are reused by other FSA.

OF-FSA. Fig. 5 provides an overview of the FSA that deals with the preposition ''of'' (OF-FSA). This FSA has one set of states in common with the BS-FSA and there are three end states indicating a successful parse. The FSA deals with structures surrounding one or two ''of's.'' There are two subtypes of patterns that we distinguish. The first and easiest pattern involves noun phrases. For example, from the sentence ''... the cytoplasmic sequestration domain of the p53 protein'' we extract the following relation ''cytoplasmic sequestration domain—of—p53 protein.''

The second pattern contains nominalizations and is more complex. Originally, we transformed all nominalizations to infinitives. For example, in the sentence ''Regulation of c-Myb activity ...'' the nominalization ''regulation'' is transformed into the infinitive ''regulate'' resulting in the following relation ''null—regulate—c-Myb activity.'' This was done so that more relations would overlap, which would be useful for later text mining and visualization. However, during initial evaluation sessions, researchers pointed out that in some cases, this is misleading. It might be, for example, that the original authors were trying to measure inhibition but did not actually find it. In this case, we need to retain the nominalization because changing it to an infinitive leads to the incorrect impression that inhibition was

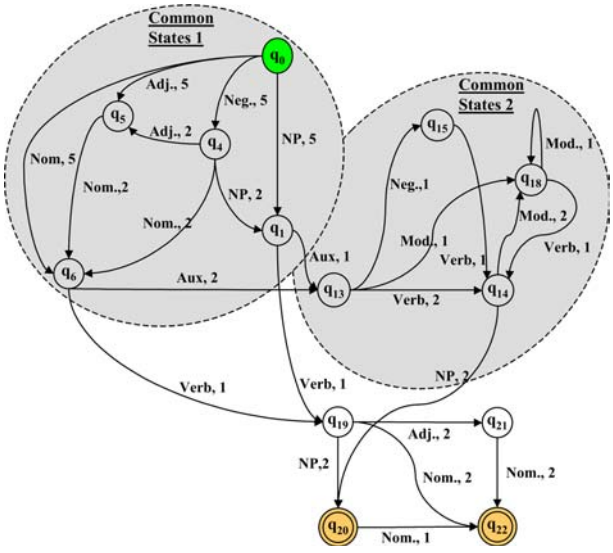


Fig. 4. Finite state automaton for basic sentences (BS-FSA: Nom., nominalization; Mod, modifier; Neg., negation; NP, noun phraser or noun; and Adj., adjective).

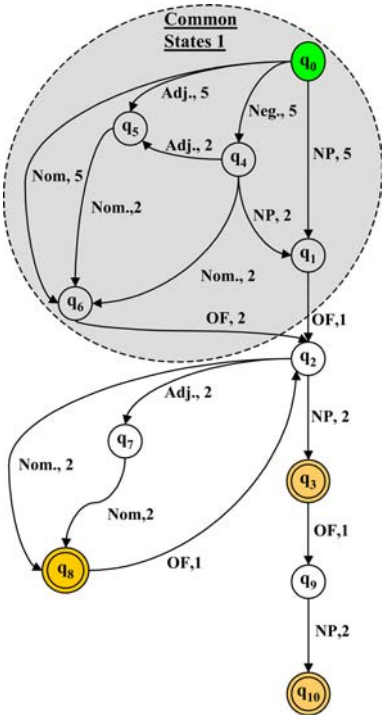


Fig. 5. Finite state automaton for the preposition ''of'' (OF-FSA: Nom., nominalization; Mod., modifier; Neg., negation; NP, noun phraser or noun; and Adj., adjective).

actually measured. To avoid misleading future users, we let the parser evaluate all verbs in a sentence. If any of the verbs indicates that the text does not discuss actual results, we do not use the infinitive representation. For this purpose, we use a list of 32 verbs, for example "anticipate," "investigate," "question." For example, from the sentence "We *propose* that E2F1 acts as a specific signal for the *induction of apoptosis* by affecting ..." we currently represent the relations as follows "induction—of—apoptosis" because of the verb "propose" that is found in the sentence. The transformation into infinitives is based on a lexical lookup of the nominalization in the UMLS Specialist Lexicon.

BY-FSA. Fig. 6 provides an overview of the FSA that deals with the preposition "by" (BY-FSA). This FSA can stand alone or it can be cascaded with the previous OF-FSA. There is one end state indicating a successful parse. When on its own, the FSA requires the presence of a verb and two noun phrases or nominalizations. It uses states common to other FSA for efficiency. For example, from the sentence "Given that E2F1 activity is stimulated by p300/CBP acetylase and ..." the relation "p300/CBP acetylase—stimulate—E2F1 activity" is extracted. When combined with the OF-FSA, it can continue from both its end states q_3 and q_8.

IN-FSA. Fig. 7 provides an overview of the FSA that deals with the preposition "in" (IN-FSA). This FSA can stand alone when there is a verb available, or it can be combined with both the OF- or BY-FSA. There is one end state indicating a successful parse. When the FSA is in stand-alone mode, the verb and the prepositions "in" are combined and stored as one connector. For example, from the sentence "These results suggest that p53 gene mutations may not occur frequently in rat bladder carcinogenesis ..." the following relation is extracted: "NOT: p53 gene mutations—(frequently) occur in—rat bladder carcinogenesis."

The IN-FSA can be combined with the OF-FSA by continuing from either of the three end states (q_3, q_8, or q_{10}). It can also be combined with the BY-FSA by continuing from end state q_{12}. The FSA can only lead to success when it ends with the preposition "in" followed by a noun phrase.

Conjunctions. The parser recognizes coordinating conjunctions. Currently, conjunctions with "and" and "or" are used that may contain any number of elements. These conjunctions are taken care of with a step-out function. When the parser encounters the start of a conjunction as indicated by "and," "or," or a comma, the FSA is halted and the parser temporarily steps out of the FSA to deal with the conjunction. It retains information about the state where the conjunction is encountered and uses heuristics to recognize valid conjunctions. Then the conjunctional constituents are stored together with the FSA state where they were encountered and the parser continues processing the FSA. This concept is illustrated in Fig. 8 and more concretely for the IN-FSA in Fig. 9.

When the parser reaches an end state successfully, the original relation is extracted together with a copy of the relation for each constituent in the conjunction. The relevant part of the copied relation is replaced with the

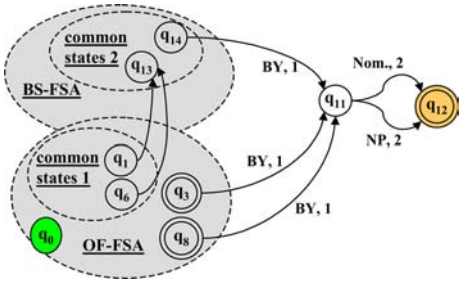


Fig. 6. Finite state automaton for the preposition "by" (BY-FSA: Nom., nominalization; Mod., modifier; Neg., negation; NP, noun phraser or noun; and Adj., adjective).

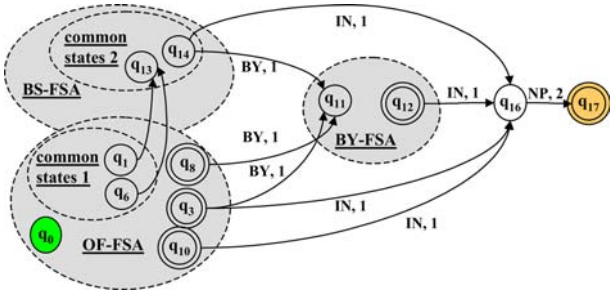


Fig. 7. Finite state automaton for the preposition "in" (IN-FSA: Nom., nominalization; Mod., modifier; Neg., negation; NP, noun phraser or noun; and Adj., adjective).

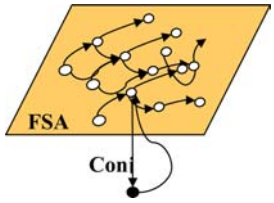


Fig. 8. Step-out function for coordinating conjunctions.

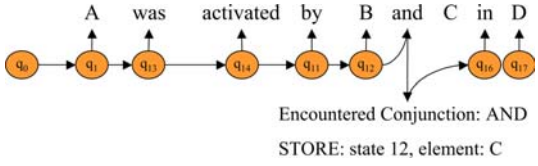


Fig. 9. Illustration of the step-out function for coordinating conjunctions.

constituent resulting in a new relation for each element in the conjunction.

For example, two sets of relations are extracted from the sentence "... induced degradation of p53 in normal thymocytes and myeloid leukemic cells." The parser first extracts the following two relations based on the IN-FSA:

- induced degradation—of—p53
- induced degradation—in—*normal thymocytes*

The second set of relations is the result of the conjunction. All information is copied, but the last element is correctly replaced, resulting in these relations:

- induced degradation—of—p53
- induced degradation—in—*myeloid leukemic cells*

To judge conjunctions we use several heuristics. They require exactly one "and" or "or" at the end and cannot be immediately followed by prepositions. All elements in a conjunction need to have the same part-of-speech and compatible semantic types. The semantic types are extracted from the UMLS Metathesaurus. To extract these types, we do a lexical lookup of the phrase, and retrieve the concept it belongs to and the semantic types of the concept. All semantic types of a concept are retrieved. We do not disambiguate the terms to assign a single semantic type. If two elements can be found in the UMLS, we consider their types compatible if both have one or more identical semantic types. This mapping captures additional errors not found based on parts-of-speech only. For example, with "breast and ovarian cancer," both "breast" and "ovarian cancer" are nouns (or noun phrases). However, the semantic types are entirely different. The reader can check this using the online UMLS Knowledge Sources (http://umlsks5.nlm.nih.gov/). These combined restrictions ensure that we capture conjunctions with complete and compatible elements. For example, in the sentence "... but not inhibitors of ERK/MAP kinase or protein kinase C ..." each constituent of the conjunctions is a complete element and two relations are extracted: "NOT: inhibitor—of—ERK/MAP kinase" and "NOT: inhibitor—of—protein kinase C."

There is no limit to the number of elements a conjunction can contain. For example, from the sentence "Immunohistochemical stains included Ber-EP4, PCNA, Ki-67, Bcl-2, p53, SM-Actin, CD31, factor XIIIa, KP-1, and CD34," 10 relations were extracted based on the same underlying pattern: "Immunohistochemical stains—include—Ber-EP4," "Immunohistochemical stains—include—PCNA," etc. At this moment, the parser captures only one conjunction per FSA.

Negation. Negation is recognized in both sets of common states (see Fig. 4) and is therefore part of each FSA. The first set of common states deals with negation that precedes a verb phrase as in for example the sentence "Yet, E2F1 does not accelerate tumor growth," the relation "NOT: E2F1 deficiency—accelerate—tumor growth" is extracted. The second set of common states deals with negation that is part of a noun phrase as in for example from the sentence "... no evidence of apoptosis ..." the relation "NOT: evidence—of—apoptosis." Currently, the parser does not handle double negation.

4. Evaluation

Three cancer researchers from the Arizona Cancer Center submitted 26 abstracts of interest to them. All 26 were parsed and 330 relations were extracted. Table 3 provides and overview. Each researcher evaluated the relations from his or her abstracts. A relation is considered correct if each component is correct, e.g., the noun phrases are complete, and if they represent the information correctly. Any incorrect component, e.g., an incomplete noun phrase, results in an incorrect relation. Additionally, even though all components can be correct, if the relation does not represent the information from the sentence correctly, if negation is missing, or if the verb infinitives are used inappropriately, the relation is scored as incorrect. Of the 330 extracted relations, 296 or 90% were correct. Five relations were correctly negated but one relation was considered incorrect because the negation was missing. We did not obtain a gold standard of all possible, relevant relations in the text from the cancer researchers because the parser focuses currently only on basic sentences and three prepositions. Therefore, we performed recall and coverage for this limited set of relations. The numbers below do not reflect the recall of all interesting relations in an abstract, but of those surrounding our target closed class words.

The researchers evaluated the relations without knowledge of the underlying FSA and so the number reported in Table 3 does not evaluate if a relation was captured by the appropriate FSA. For example, if a relation was captured by the OF-FSA but should have been captured by the BY-FSA, the researchers considered the relation correct as long as it correctly represented the information in the sentence. To better understand how each FSA contributed to the results, we calculated precision and recall per FSA and for the conjunctions. This evaluation is presented in the following sections. It is based on the evaluation of the

Table 3
Overview of the abstracts and relations

	Total	Average per abstract
Abstracts	26	
Sentences	237	9
Extracted relations	330	13
Correct relations	296 (90%)	11

researchers of the correctness of relations but with the additional condition that each relation needs to be completely extracted by the correct FSA to be considered correct.

4.1. Precision of the FSA

Precision was calculated by dividing the number of correctly and completely extracted relations by the total number of extracted relations. The "correctly extracted relations" are those relations considered correct by the researchers, as described above, but with the additional restriction that it needs to be completely extracted by the appropriate FSA. This is a more strict evaluation.

Precision =# correctly and completely extracted relations/# total extracted relations

Table 4 provides an overview of the precision of the relations. There were 267 relations extracted from the abstracts (excludes the conjunctional copies); 179 were extracted completely and correctly resulting in 89% precision. A closer look at all errors revealed that nine errors (38%) were due to incorrect noun phrases in the relation and two errors (8%) were due to an incorrect transformation of a nominalization to a verb infinitive. Precision was highest for the OF-FSA (92%) and lowest for the basic sentences (53%).

4.2. Recall of the FSA

We calculated recall in a similar manner as precision: the number of correctly and completely extracted relations divided by the total number of relations available in the text:

Recall = # correctly and completely extracted relations/# total relations

The "total relations" were counted by doing a manual check of all sentences in the abstracts. When a sentence contains the required prepositions for a particular pattern and the distance between the elements was less or equal to the maximally allowed distance, the relation was counted as "required." When a relation contained a conjunction, it was only counted once, since we evaluate conjunctions separately in the following section.

Only those relations that could have been captured in the text with the current FSA were considered. Table 5 shows an overview of the recall of relations per FSA. Overall, 62% of the patterns were correctly and completely extracted. As with precision, conjunctional copies are not considered here. The highest recall was found for the OF-FSA, 71% recall, and the BY-FSA, 63% recall. Recall was lowest for the IN-FSA (30%) where a relation was considered missing when the noun phrase introduced by "in" was missing. These relations were often extracted by another FSA but considered incorrect here.

4.3. Precision and recall of conjunctions

To calculate precision and recall of conjunctions, we counted each relation in the text where a conjunction was part of the FSA pattern. Conjunctions where the elements needed recombination, e.g., "breast and ovarian cancer," are not counted since we explicitly avoid them. A conjunction is considered to be completely and correctly extracted if each element is placed in the correct FSA relation. If any of the elements is incorrect, if the relation is incorrect, or if any element is missing from the copied relation, e.g., a negation, we consider this to be an incorrect conjunction resulting in lower precision. If any copy is missing, we consider this a missed conjunction resulting in lower recall.

Tables 6 and 7 provide an overview of the results. There were 30 valid conjunctions in the abstracts. Of these, 12 were correctly and completely extracted. A conjunction was either correctly extracted (100% precision) or it was ignored. This results in a few selective relations being added to the result set without introducing any new errors.

Table 5
Recall of the relations

FSA	Total correct	Total relations	Recall (%)
BS-FSA	8	23	35
OF-FSA	145	203	71
BY-FSA	15	24	63
IN-FSA	11	37	30
All	179	287	62

Table 4
Precision of the relations

FSA	Total correct	Total extracted	Precision (%)
BS-FSA	8	15	53
OF-FSA	145	157	92
BY-FSA	15	17	88
IN-FSA	11	13	85
All	179	202	89

Table 6
Precision of the conjunctions

FSA	Total correct	Total extracted	Precision (%)
BS-FSA	1	1	100
OF-FSA	10	10	100
BY-FSA	0	0	—
IN-FSA	1	1	100
All	12	12	100

Table 7
Recall of the conjunctions

FSA	Total correct	Total conjunctions	Recall (%)
BS-FSA	1	1	100
OF-FSA	10	22	45
BY-FSA	0	1	0
IN-FSA	1	6	16
All	12	30	40

Table 8
Coverage of prepositions and conjunctions

26 Abstracts	Prepositions		
	of	by	in
Total in abstracts	257	66	130
Correctly captured	197	19	18
Coverage	77%	29%	14%

4.4. Coverage of relations

To learn the coverage of the combined FSA patterns, we counted the three prepositions in the abstracts. All occurrences of "by," "of," and "in" were counted with the following exceptions: "in addition," "in view," "in this report," "in order," and "in contrast" which appeared usually in the beginning of the sentence and are explicitly avoided by the parser because they result in irrelevant relations. In addition, "in" was not counted as a preposition when encountered as part of "in vivo" and "in vitro."

The results are summarized in Table 8. We considered the prepositions captured when it was part of a correct relation. Seventy seven percent of all "of" prepositions, 29% of all "by" prepositions, and 14% of all "in" prepositions were correctly captured by the FSA. These numbers indicate that the OF-FSA is relatively complete for this type of biomedical text. The BY-FSA and IN-FSA cover a smaller portion of the available structures.

5. Conclusion

In this paper, we presented a shallow parser based on closed-class English words to efficiently capture relations between noun phrases in biomedical text. Cascaded FSA model the relations resulting in time-efficient processing. Relations are not limited to certain words, e.g., proteins, or certain verbs, e.g., activate, and can contain up to five elements: the left-hand side (LHS) and right-hand side (RHS) of a relation, a connector which binds the LHS to the RHS, a modifier, and negation. We tested our approach on 26 abstracts of interest to three cancer researchers who subsequently evaluated the relations extracted by the parser. On average, there were 11 correct relations extracted per abstract, 296 in total, with 90% precision.

The precision (90%) we achieved makes our parsing approach comparable to the best. Others report precision ranging from 60 to 96% [29,30,32]. Comparing recall with others is harder, since different types of relations are extracted and it is often hard to find exact numbers. In general, we believe our parser extracts more relations per abstract because we do not limit the relations to specific verbs or specific entities. GENIES, for example, extracted 27 relations between biological molecules, 19 of which were unique, from one full text article [32]. Ng and Wong [56] found 16 unique protein–protein interactions in 26 Medline abstracts. Thomas et al. [29] estimate that there exists one relevant relation in half of their 2565 Medline abstracts and report a recall of 80% for their best sample.

The parser has limitations that need to be addressed in the near future. First of all, the coverage of the prepositions provides us with a guideline on where to focus expansion efforts. We plan to complete the patterns for "by" and "in," before moving on to other prepositions. In addition, a more general, complete, and linguistically sound approach towards conjunctions needs to be used. Currently, only one conjunction can be dealt with per FSA. Finally, our study is limited because no complete gold standard for all relevant relations was available for our test set. Instead, we approximated recall of the patterns. We counted all the occurrence of the prepositions. The counts of the prepositions in the order covered by the parser were used to calculate recall. The count of all occurrences was used to calculate coverage.

6. Future directions

In the near future, we plan to improve and expand the parser. Initially, we will add more patterns for the same prepositions, later we will add more prepositions. We will look further into the differences between relations that represent actual results or not. We also plan to add a module that can combine these structures into more complex hierarchies. This would be necessary to deal with structures introduced by, for example, subordinating conjunctions.

Currently, we are collaborating with cancer researchers from the Arizona Cancer Center who are interested in the p53 gene. As of August 2002, there were 23,265 abstracts in Medline that contained the keyword p53 in either the title or the abstract text. On a computer with a 1-GHz processor and 392-MB RAM, the parser processed seven abstracts per second and this p53-collection was processed in about 1 h demonstrating the scalability of our approach. All relations and original

abstracts are stored in our Genescene knowledge base. We plan to add meta-information about the extracted relations to Genescene and hope to achieve this by tagging all elements with information from the UMLS, the Gene Ontology, and the Human Genome Nomenclature. All three knowledge sources are currently integrated in Genescene and used to tag each element. We are working on algorithms to choose a unique tag for each element instead of multiple tags. This will allow us to label individual elements with relevant tags such as gene, disease, or patient group.

Our goal is to visualize the relations extracted from each collection in a semantically rich map that researchers can browse. Because of this goal, we focused on finding semantically rich but precisely extracted relations. By tagging all elements, users will be able to limit the map to only those parts of interests, e.g., only relations between genes and diseases. An online demo is available at http://ai.bpa.arizona.edu/go/GeneScene.

Acknowledgments

This project was supported by the following grant: NIH/NLM, 1 R33 LM07299-01, 2002-2005, "Genescene: a Toolkit for Gene Pathway Analysis". We thank Ryan Falsey and Kerri Kislin from the Arizona Cancer Center for their suggestion, ideas, and evaluations. We also thank the National Library of Medicine, the Gene Ontology Consortium, and the Human Genome Nomenclature Committee for making the UMLS, GO, and HUGO freely available to researchers.

References

[1] Maojo V, Iakovidis I, Martin-Sanchez F, Crespo J, Kulikowski C. Medical information and bioinformatics: european efforts to facilitate synergy. J Biomed Inform 2001;34:423–7.
[2] McCray AT, Aronson AR, Browne AC, Rindflesch TC, Razi A, Srinivasan S. UMLS knowledge for biomedical language processing. Bull Med Libr Assoc 1993;81(2):184–94.
[3] Hersh WR, Campbell EM, Malveau SE. Assessing the feasibility of large-scale natural language processing in a corpus of ordinary medical records: a lexical analysis. In: Proceedings of the 1997 AMIA Annual Symposium; 1997. p. 580–84.
[4] Aronson AR. effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In: AMIA Symposium; 2001. p. 17–21.
[5] Humphreys BL, McCray AT, Cheh ML. Evaluating the coverage of controlled health data terminologies: report on the results of the NLM/AHCPR large scale vocabulary test. J Am Med Inform Assoc 1997;4(6):484–500.
[6] Wain HM, Lush M, Ducluzeau F, Povey S. Genew: the human gene nomenclature database. Nucleic Acids Res 2002;30(1):169–71.
[7] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene ontology: tool for the unification of biology. Nat Genet 2000;25:25–9.
[8] Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. Genome Research 2001;11(8):1425–33.
[9] Ohta T, Tateisi Y, Kim J-D. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In: Human Language Technology Conference, San Diego, California, USA; 2002.
[10] Weischedel R, Meteer M, Schwartz R, Ramshaw L, Palmucci J. Coping with ambiguity and unknown words through probabilistic models. Comput Linguist 1993;19(2):359–82.
[11] Hindle D. Deterministic parsing of syntactic non-fluencies. In: 21st Annual Meeting of the Association for Computational Linguistics; 1983. p. 123–28.
[12] Church KW. A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the Second Conference on Applied Natural Language Processing; 1988. p. 136–43.
[13] Vourtilainen A, Padro L. Developing a hybrid NP parser. In: Fifth Conference on Applied Natural Language Processing; 1997. p. 80–87.
[14] McDonald DD. Robust partial parsing through incremental, multi-algorithm processing. In: Jacobs PS, editor. Text-based intelligent systems. 1992. p. 83–99.
[15] Tolle KM, Chen H. Comparing noun phrasing techniques for use with medical digital library tools. J Am Soc Inform Syst 2000;51(4):352–70.
[16] Hersh W, Mailhot M, Arnott-Smith C, Lowe H. Selective automated indexing of findings and diagnoses in radiology reports. J Biomed Inform 2001;34:262–73.
[17] Rindflesch TC, Hunter L, Aronson AR. Mining molecular binding terminology from biomedical text, In: Amia Fall Symposium; 1999, p. 127–31.
[18] Raychaudhuri S, Chang JT, Sutphin PD, Altman RB. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. Genome Res 2002;12:203–14.
[19] Kazama Ji, Maino T, Ohta Y, Tsujii Ji. Tuning support vector machines for biomedical named entity recognition. In: Association for Computation Linguistics Workshop on Natural Language Processing in the Biomedical Domain. Philadelphia: ACL; 2002.
[20] Fukuda K, Tsunoda T, Tamura A, Takagi T. Toward information extraction: identifying protein names from biological papers. In: Pacific Symposium on Biocomputing; 1998. p. 705–16.
[21] Cohen KB, Dolbey AE, Acquaash-Mensah GK, Hunter L. Contrast and variability in gene names. In: Workshop on Natural Language Processing in the Biomedical Domain: Association for Computational Linguistics; 2002. p. 14–20.
[22] Hatzivassiloglou V, Duboué PA. Disambiguating proteins, genes, and RNA in text: a machine learning approach. Bioinformatics 2001;1(1):1–10.
[23] Krauthammer M, Rzhetsky A, Morozov P, Friedman C. Using BLAST for identifying gene and protein names in journal articles. Gene 2000;259:245–52.
[24] Proux D, Rechenmann F, Julliard L, Pillet V, Jacq B. Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. In: Ninth Workshop on Genome Informatics; 1998. p. 72–80.
[25] Liu H, Lussier YA, Friedman C. Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method. J Biomed Inform 2001;34:249–61.
[26] Jenssen T-K, Laegreid A, Komorowski J, Hovig E. A literature network of human genes for high-throughput analysis of gene expression. Nat Genet 2001;28:21–8.
[27] Blaschke C, Valencia A. Can bibliographic pointers for known biological data be found automatically? Protein interactions as a case study. Comp Funct Genom 2001;2:196–206.
[28] Sekimisu T, Park HS, Tsujii Ji. Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. Genome Inform 1998:62–71.

[29] Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M. Automatic extraction of protein interactions from scientific abstracts. In: Pacific Symposium on Biocomputing; 2000. p. 538–49.
[30] Pustejovsky J, Castaño J, Zhang J, Kotecki M, Cochran B. Robust relational parsing over biomedical literature: extracting inhibit relations. In: Pacific Symposium on Biocomputing; 2002. p. 362–73.
[31] Gaizauskas R, Demetriou G, Artymiuk PJ, Willett P. Protein structures and information extraction from biological texts: the PASTA system. Bioinformatics 2003;19(1):135–43.
[32] Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 2001;17(Suppl 1):S74–82.
[33] Friedman C, Hripcsak G. Evaluating natural language processors in the clinical domain. Methods of Information in Medicine 1998;37:334–44.
[34] Barrows RC, Busuioc M, Friedman C. Limited parsing of notational text visit notes: ad-hoc vs. NLP approaches. In: AMIA 2000 Symposium; 2000. p. 51–5.
[35] Friedman C. A broad-coverage natural language processing system. In: AMIA 2000 Annual Symposium; 2000.
[36] Pullum GK, Huddleston R. Prepositions and preposition phrases. In: Pullum GK, editor. The Cambridge grammar of the English language. Cambridge, UK: Cambridge University Press; 2002.
[37] Manning CD, Schütze H. Foundations of statistical natural language processing. Cambridge, MA: MIT Press; 2001.
[38] Jolly J. Prepositional analysis within the framework of role and reference grammar. New York: Peter Lang Publishing; 1991.
[39] Ratnaparkhi A, Reynar J, Roukos S. A maximum entropy model for prepositional phrase attachment. In: ARPA Human Language Technology Workshop; 1994. p. 250–55.
[40] Brill E, Resnik P. A rule-based approach to prepositional phrase attachment disambiguation. In: COLING; 1994.
[41] Ratnaparkhi A. Statistical models for unsupervised prepositional phrase attachment. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics; 1998.
[42] Abney S, Schapire RE, Singer Y. Boosting applied to tagging and PP attachment. In: Empirical Methods in Natural Language Processing and Very Large Corpora; 1999.
[43] Leroy G, Chen H. Filling preposition-based templates to capture information from medical abstracts. In: Pacific Symposium on Biocomputing, January, Kauai; 2002. p. 350–61.
[44] Tottie G. Negation in English speech and writing: a study in variation. New York: Academic Press; 1991.
[45] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 2001;34:301–10.
[46] Friedman C, Alderson PO, Austin JM, Cimino JJ, Johnson SB. A general natural-language text processor for clinical radiology. J Am Med Inform Assoc 1994;1(2):161–74.
[47] Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents. J Am Med Inform Assoc 2001;8:598–609.
[48] Jurafsky D, Martin JH. Regular expressions and automata. In: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, NJ: Prentice-Hall; 2000. p. 21–56.
[49] Roche E, Schabes Y. Deterministic part-of-speech tagging with finite state transducers. In: Schabes Y, editor. Finite-state language processing. Cambridge, MA: The MIT Press; 1997.
[50] Jurafsky D, Martin JH. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, NJ: Prentice-Hall; 2000.
[51] Kokkinakis D, Johansson-Kokkinakis S. A cascaded finite-state parser for syntactic analysis of swedish. In: The 9th EACL, Bergen, Norway; 1999.
[52] Grefenstette G. Light parsing as finite-state filtering. In: Workshop on Extended Finite State Models of Language (ECAI'96), Budapest, Hungary; 1996.
[53] Abney S. Partial parsing via finite-state cascades. In: 8th European Summer School in Logic, Language and Information—Workshop on Robust Parsing: Prague, Czech Republic; 1996. p. 8–15.
[54] Van Delden S., Gomez F. Combining finite state automata and a greedy learning algorithm to determine the syntactic roles of commas. In: 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02); 2002.
[55] Roche E. Parsing with finite state transducers. In: Schabes Y, editor. Finite-state language processing. Cambridge, MA: The MIT Press; 1997. p. 241–81.
[56] Ng S-K, Wong M. Toward routine automatic pathway discovery from on-line scientific text abstracts. Genome Inform 1999;10:104–12.